Bayesian Hierarchical Modeling

David Draper

Department of
Applied Mathematics and Statistics
University of California, Santa Cruz

draper@ams.ucsc.edu

http://www.ams.ucsc.edu/~draper/

Draft 7 (January 2005): revised Contents and Preface, **Chapter 1**, **Chapter 2**, *and revised References. Comments welcome.*

This book was typeset by the author using a PostScript (Adobe Systems, Inc.) based phototypesetter. The figures were generated in PostScript using the S+ data analysis language (Becker et al., 1988), and were directly incorporated into the typeset document. The text was formatted using the LaTeX language (Lamport, 1994), a version of TeX (Knuth, 1984).

TO ANDREA

Contents

Preface

This book provides an introduction to the formulation, fitting, and checking of **hierarchical** or **multi-level** models, from the Bayesian point of view. Hierarchical models (HMs) arise frequently in five main kinds of applications:

- HMs are common in fields such as health and education, in which data—both outcomes and predictors—are often gathered in a *nested* or *hierarchical* fashion: for example, patients within hospitals, or students within classrooms within schools. HMs are thus also ideally suited to the wide range of applications in government and business in which single- or multi-stage *cluster samples* are routinely drawn, and offer a unified approach to the analysis of **random-effects (variance-components)** and **mixed models**.
- A different kind of nested data arises in **meta-analysis** in, e.g., medicine and the social sciences. In this setting the goal is *combining information* from a number of studies of essentially the same phenomenon, to produce more accurate inferences and predictions than those available from any single study. Here the data structure is subjects within studies, and as in the clustered case above there will generally be predictors available at both the subject and study levels.
- When individuals—in medicine, for instance—are sampled cross-sectionally but then studied longitudinally, with outcomes observed at multiple time points for each person, a hierarchical data structure of the type studied in **repeated-measures** or **growth curve** analyses arises, with the readings at different time points nested within person.
- For simplicity people often try to model data as (conditionally) IID at a fairly high level of aggregation—for instance, by pretending that all the subjects in a sampling experiment are drawn homogeneously from a single population. In fact, heterogene-

ity is often the rule rather than the exception, and frequently the available predictor variables do not "explain" this heterogeneity sufficiently. With recent computational advances it is becoming increasingly straightforward to at least *describe* such heterogeneity with **mixture models** that employ **latent variables** (unobserved predictors) in a hierarchical structure. Examples include **density estimation** with an unknown number of sub-populations mixed together and **Bayesian nonparametric modeling**, in which people work with distributions whose sample spaces are themselves sets of distributions instead of (say) real numbers.

- Hierarchical modeling also provides a natural way to treat issues of **model selection** and **model uncertainty** with all types of data, not just cluster samples or repeated measures outcomes. For example, in regression, if the data appear to exhibit residual variation that changes with the predictors, you can *expand the model* that assumes constant variation, by embedding it hierarchically in a family of models that span a variety of assumptions about residual variation. In this way, instead of having to choose one of these models and risk making the wrong choice, you can work with several models at once, weighting them in proportion to their plausibility given the data.

In studying HMs there are two kinds of technical issues that also arise: fully Bayesian computation in HMs requires the use of simulation methods such as those based on **Markov Chain Monte Carlo (MCMC)** ideas, and—as usual with any class of statistical models—there are questions of **model diagnostics**.

Plan of the book. In the chapters below I describe the principles of Bayesian hierarchical modeling, with emphasis on practical rather than theoretical issues, and I illustrate these principles with analyses of real data drawn from case studies. The material is intended for applied statisticians with an interest in learning more about hierarchical models in general, and the Bayesian analysis of such models in particular. The field of study examined here is surprisingly wide, touching on topics in numerical analysis, high-dimensional integration, and measures on function space (on the mathematical side), the meaning of uncertainty and probability (in philosophy and statistics), and practical issues in Markov chains, time series, and modern nonparametric analysis.

The nine chapters cover the five application areas mentioned above, together with an introductory chapter on Bayesian modeling, one chapter each on MCMC and model diagnostics, and a concluding chapter with discussion and suggestions for future research. An appendix reviews standard probability distributions useful in Bayesian work, and another provides computing details in the environments I used to write the book: the statistical computing and graphics package `S+`, the Gibbs sampling package `BUGS`, the multi-level modeling package `MLwiN`, the symbolic computing package `Maple`, and the high-level programming language `C`.

An understanding of probability at the level typically required for a master's degree in statistics provides ample mathematical background. I have taught subsets of this material successfully to groups including British final-year undergraduates, American PhD students, and PhD-level researchers enrolled in short courses, and the book has also proven useful for self-study by researchers and graduate students in a variety of disciplines (including statistics).

No previous experience with Bayesian methods is needed—all relevant ideas are covered in a self-contained fashion. If you already know a fair bit about Bayes you can move through Chapter 1 briskly, although there are philosophical and practical issues of potential interest even to seasoned Bayesians there. If you are new to Bayes, a good way to read this book is in conjunction with one or both of the following excellent publications: the Bayesian text by Gelman et al. (1995), and the monograph on MCMC by Gilks et al. (1996) (although the latter is at a more advanced level than the former). A supplementary and complementary perspective on many of the issues covered here can also be obtained by doing some reading in parallel in the excellent book by Carlin and Louis (1996).

Some style and layout conventions to be aware of in the chapters that follow:

- I like to teach and talk about research ideas informally, and the book reflects this. I have tried to write as if you and I were having an extended conversation on the topics covered here. This is natural in a book on applications of the Bayesian approach to probability, and has various advantages, but one possible disadvantage is that the scope of agreement in the statistics community with statements I make may not be immediately clear. So here is a dictionary: sentences including phrases like "You

can show that" and "Evidently" are meant to be expressions of mathematical fact; phrases like "Most people believe that" signal general unanimity (in my view) among (Bayesian) statisticians on the point I'm covering; and phrases like "It seems to me that" precede a personal opinion of mine, which may or may not be shared by other statisticians.

- I am writing in LaTeX, and I don't like LaTeX's subsection layout, so one-line text boxes act as subsection headings. Multi-line text boxes, in contrast, bring emphasis to definitions, theorems, and summaries of important points.
- The book is dotted with blocks of text that begin **NB**—these highlight things like general notational conventions and pitfalls to be avoided in implementing the ideas I'm discussing at that point.
- **Bold font** is generally reserved for the first appearance of important technical terms, and *italics* signal items of particular emphasis.
- I have tried to write for a fairly diverse audience in terms of mathematical and statistical background. One of the main devices for (I hope) achieving this fairly smoothly is *footnotes*[0], which are often too long to be at the bottom of the page where they belong, so I have collected them at the end of each chapter. The naming convention is that, for instance, note[6] in Chapter 3 will be found as item 3.6 in the Notes section of that chapter. In general, the footnotes supplement the main text by adding historical details, additional mathematical formalism, notices of nonstandard terminology, and the like. The intent is that if you are new to much of this material, you can skip (many or all of) the notes on first reading if you want; whereas if you are fairly experienced in the topics covered here, or you want to dig a bit deeper, you may find that the notes enrich the material and suggest directions for further reading.
- I also offer a somewhat eclectic variety of problems in each chapter: some are data-analytic, others somewhat more theoretical, and they vary widely in difficulty. Problems that use material in the notes begin with the symbol ($\mathcal{N}n$), where n refers to the chapter in which the relevant notes may be found. To get the most out of the material, I recommend not only working many or all of the problems but also programming up most or all of

the examples and case studies to see if you get results similar to mine.

I am grateful to Bill Browne, Ryan Cheal, Dimitris Fouskakis, David Freedman, Andrew Gelman, Sander Greenland, Merilee Hurn, Dennis Lindley, Nick Longford, David Madigan, Colin Mallows, Michael Seltzer, and David Williams for comments on earlier versions of this material, and to the UK Engineering and Physical Sciences Research Council, the European Commission, the University of Bath (UK), and the University of California, Santa Cruz for support. Membership on this list does not imply agreement with the ideas expressed here, nor are any of these people or institutions responsible for any errors that may be present.

Santa Cruz, California David Draper
January 2005

CHAPTER 1

Introduction to Bayesian modeling

1.1 Quantification of uncertainty about observables

Case study 1.1: *Hospital-specific prediction of mortality rates.* Let's say you are interested in measuring the *quality of care* (e.g., Kahn et al., 1990) offered by one particular hospital. I am thinking of the Royal United Hospital (RUH) in Bath, England, where I work; you will probably have a different hospital in mind.

As part of this you decide to examine the medical records of all patients treated at the RUH in one particular time window, say January 1996–December 1999, for one particular medical condition for which there is a strong *process-outcome link*[1], say acute myocardial infarction (AMI; heart attack). In the time window you're interested in there will be about $n = 400$ AMI patients at the RUH.

To keep things simple let's ignore process for the moment and focus here on one particular outcome: *death status (mortality)* as of 30 days from hospital admission, coded 1 for dead and 0 for alive. (In addition to process this will also depend on the *sickness at admission* of the AMI patients, but let's ignore that initially too.) From the vantage point of December 1995, say, what may be said about the roughly 400 1's and 0's you will observe in 1996–99?

The meaning of probability. You are definitely *uncertain* about the 0–1 death outcomes $Y_1, \ldots, Y_n$ before you observe any of them. **Probability** is supposed to be the part of mathematics concerned with quantifying uncertainty[2]; how can probability be used here?

Consider a description A of some aspect of something about which you are uncertain. (Here, for example, A could be $(Y_i = 1) =$ {patient i will die} for some i.) Three main approaches to endowing probabilities with real-world meaning have so far been developed (e.g., Oakes, 1986[3], Hacking, 1975): **classical, frequentist** and **Bayesian**.

- **Classical**: Enumerate *elemental outcomes* (EOs) in a way that makes them *equipossible* on the basis of symmetry considerations, and compute

$$P_C(A) \equiv \frac{n_A}{n} = \frac{\text{number of EOs favorable to } A}{\text{total number of EOs}}. \tag{1.1}$$

- **Frequentist**: Restrict attention to *attributes* A of *events* (phenomena that are inherently repeatable under "identical" conditions) and define

$$P_F(A) \equiv \lim_{n\to\infty} \frac{\#\text{ of repetitions in which } A \text{ occurs}}{n}. \tag{1.2}$$

- **Bayesian**: Imagine betting with someone about the truth of a *proposition* A (propositions can be anything—not just repeatable phenomena—whose truth value is not (yet) known), and ask yourself what odds $(O_A|\mathcal{B}_{\text{you}})$ you would need to give or receive in order that you judge the bet fair, where $\mathcal{B}_{\text{you}}$ represents your knowledge and beliefs relevant to the assessment of the odds; then (for you)

$$P_{B:\text{you}}(A) \equiv P_B(A|\mathcal{B}_{\text{you}}) \equiv \frac{(O_A|\mathcal{B}_{\text{you}})}{1+(O_A|\mathcal{B}_{\text{you}})}. \tag{1.3}$$

> **NB** Some notational conventions: (1) In what follows I will usually just write $\mathcal{B}$ instead of $\mathcal{B}_{\text{you}}$; (2) When it is clear from context that I am talking about a Bayesian probability, I will generally drop the B in P_B; and (3) For brevity I will sometimes omit the explicit conditioning on your beliefs $\mathcal{B}$ in the notation. This should always be regarded as present, even when not actually printed in the conditional probability expressions.

Each of these probability definitions has general advantages and disadvantages:

- **Classical**
  - *Plus*: When relevant, this definition is simple—most people are first taught classical probability, with toy examples like idealized coin-tossing and drawing balls from urns.
  - *Minus*: The only way to define "equipossible" without a circular appeal to probability is through the *principle of insufficient reason*—you judge EOs equipossible if you have no

grounds (empirical, logical, or symmetrical) for favoring one over another—but this leads to paradoxes (for instance, the assertion of equal uncertainty is not invariant to the choice of scale on which it is asserted[4]).

- **Frequentist**
  - *Plus*: Mathematical analysis with this approach is relatively tractable, which helps to explain the widespread use of frequentist probability in mathematical statistics over the last 100 years.
  - *Minus*: But the frequentist definition only applies to inherently repeatable events: for example, P_F(Al Gore will be elected president of the United States in 2000) is (strictly speaking) undefined.
- **Bayesian**
  - *Plus*: All forms of uncertainty are inherently quantifiable with this approach.
  - *Minus*: There is no guarantee that the answer you get by querying yourself about betting odds will retrospectively be seen by you or others as "good" (but how should the quality of an uncertainty assessment itself be assessed?).

Application to mortality prediction. Suppose for the moment that you did in fact have a variety of process and admission sickness variables available for a large collection $\mathcal{P}$ of AMI patients, and you were trying to assess the probability that a particular patient—let's call her S—with a given process and admission sickness profile will die within 30 days of admission. How would the three definitions above be applied to this assessment?

If you think about how you would try to quantify this patient's risk of dying, you will see that all three approaches require you to make judgments about the *similarity* of this patient to other patients. The English statistician and geneticist Fisher defined the **recognizable subpopulation** $\mathcal{P}_S$ to which this patient belongs as his way of coming to grips with similarity judgments:

> **Definition** (Fisher, 1956): The *recognizable subpopulation* $\mathcal{P}_S$ for patient S is the smallest subset to which she belongs for which the AMI mortality rate differs from that in the rest of $\mathcal{P}$ by an amount you judge as significant in a practical sense.

Within $\mathcal{P}_S$ you regard the risk of dying as close enough to constant that the differences aren't worth bothering over, but the differences between mortality rates in $\mathcal{P}_S$ and its complement matter to you. I will address below how you would go about identifying $\mathcal{P}_S$ in practice.

Taking it as given that $\mathcal{P}_S$ has been established, as a *classicist* you would then (a) use Fisher's definition to establish equipossibility within $\mathcal{P}_S$, (b) count $n_A =$ (number of deaths in $\mathcal{P}_S$) and $n =$ (total number of people in $\mathcal{P}_S$), and (c) compute $P_C(A) = \frac{n_A}{n}$.

As a *frequentist*, to bring in the idea of repeating something under "identical" conditions, you would have to (a) equate $P(A)$ to P(a person chosen at random **(IID)** from $\mathcal{P}_S$ dies), (b) imagine repeating this random sampling indefinitely, and (c) conclude that the limiting value of the relative frequency of mortality in these repetitions would be $P_F(A) = \frac{n_A}{n}$. Notice that strictly speaking you can't talk about P_F(*this* patient will die)—you have to imagine embedding this patient in a repeatable sequence and settle for saying something about the *sequence*.

As a *Bayesian*, with the information given here you would regard this patient as **exchangeable** with all other patients in $\mathcal{P}_S$—meaning informally that you judge yourself equally uncertain about mortality for all the patients in this set—and this judgment, together with the axioms of **coherence** (a kind of internal consistency requirement; see Note 1.16), would also yield $P_{B:\text{you}}(A) = \frac{n_A}{n}$ (although I have not yet said why this is so). I will look at exchangeability and coherence in more detail below.

Note that with the same information base the three approaches in this case have led to the same answer, although the *meaning* of that answer depends on the approach. For example, frequentist probability describes the *process* of observing a repeatable event whereas Bayesian probability is an attempt to quantify your uncertainty about something, repeatable or not.

Subjectivity and "objectivity." The classical and frequentist approaches have sometimes been called **"objective,"** whereas the Bayesian approach is clearly **subjective** or **judgmental**. I would argue, however, that in interesting applied problems of realistic complexity, the judgment of *similarity* (equipossibility, IID, exchangeability) that is evidently central to all three theories makes them all subjective in practice.

Imagine, for instance, that you were given data on death status

in a large group of AMI patients, along with many variables that might or might not be relevant to predicting their mortality, and asked to identify $\mathcal{P}_{\mathcal{S}}$. You might build a generalized linear model to estimate P(death within 30 days) from the available predictors. But in building this model you would make many judgment calls, for example the choice of link function (logit versus complementary log-log, say) and the "best" subset of predictors to include. The result could easily be considerable variation in the estimates of P(death) obtained by you and other reasonable analysts working independently[5], and the differences between the answers obtained in this way come entirely from the exercise of modeling judgment.

Thus the assessment of complicated probabilities is *inherently subjective*. With this in mind attention in all three approaches should perhaps shift away from trying to achieve "objectivity" toward the explicit statement of the assumptions and judgments made in forming probability assessments, so that consumers of these assessments may judge their plausibility[6].

Frequentist modeling. I will focus on the approaches with the most widespread usage—*frequentist* and *Bayesian*—in the rest of the book. How, for instance, can the frequentist definition of probability be applied to the hospital mortality problem?

As a frequentist, to use probability to quantify your uncertainty about the 1's and 0's, you have to think of them as either literally a *random sample* or *like* a random sample from some population, either hypothetical or actual.

- An example of a hypothetical population would be all AMI patients who *might have* come to the RUH in 1996–99 if the world had turned out differently in some (unspecified) ways.
- Some actual populations: (1) Assuming sufficient time-homogeneity in all relevant factors, you could try to argue that the collection of all 400 AMI patients at the RUH from 1996–99 is *like* a random sample of size 400 from the population of all AMI patients at the RUH from (say) 1993–2002, even though in fact it is a kind of *time-cluster* sample in which you got everybody from 1996–99 and nobody from 1993–95 or 2000–02; or (2) Assuming the RUH to be representative of some broader collection of hospitals in England and ignoring intracluster correlation, you could try to argue that a cluster sample of all 400 AMI patients from the RUH was *like* a simple random sample of 400 AMI patients from this larger collection of hospitals.

None of these options is, shall we say, entirely compelling[7].

If you are willing to pretend the data are like a sample from some population, you could then regard the 400 1's and 0's at the RUH as realizations of random variables and begin to think about a **model**, for example

$$Y_i \overset{\text{indep}}{\sim} \mathrm{B}(\theta_i), \quad i = 1, \ldots, n, \tag{1.4}$$

where $B(\cdot)$ denotes the Bernoulli distribution. (Appendix 1 contains a summary of the distributions used in this book.) In the absence of any sickness or process information, however, you would probably have to treat the 1's and 0's as homogeneous and work with the simpler model

$$Y_i \overset{\text{IID}}{\sim} \mathrm{B}(\theta), \quad i = 1, \ldots, n. \tag{1.5}$$

Interest would then focus on **inference** about the **parameter** θ, the "underlying death rate": if θ were unusually high, that would be *prima facie* evidence of a possible quality of care problem at the RUH[8].

Bayesian modeling. As a Bayesian in this situation, your job is to quantify your uncertainty about the 400 binary **observables** you will begin to see starting in 1996—in other words, your initial modeling task is **predictive** rather than inferential. There is no samples-and-populations story in this approach, but probability and random variables arise in a different way: quantifying your uncertainty (for the purpose of betting with someone about some aspect of the 1's and 0's, say) requires **eliciting** from yourself a joint probability distribution that **accurately** captures your judgments about what you will see[9]:

$$P_{B:\text{you}}(Y_1 = y_1, \ldots, Y_n = y_n). \tag{1.6}$$

Notice as before that in the frequentist approach the random variables describe the *process* of observing a repeatable event (the "random sampling" appealed to here), whereas in the Bayesian approach you use random variables to quantify *your uncertainty about observables you haven't seen yet*[10].

I will argue later (Section 1.5) that the concept of probabilistic *accuracy* has two components: you want your uncertainty assessments to be both *internally* and *externally* consistent, which corresponds to the ideas of **coherence** and **calibration**, respectively.

1.2 Discrete outcomes: Exchangeability

Eliciting a 400-dimensional distribution doesn't sound easy—major simplification is evidently needed. In this case, and many others, this is provided by **exchangeability** considerations. If (as in the frequentist approach) you have no relevant information that distinguishes one AMI patient from another, your uncertainty about the 400 1's and 0's is *symmetric*, in the sense that a random permutation of the *order* in which the 1's and 0's were labeled from 1 to 400 would leave your uncertainty about them unchanged. The Italian statistician de Finetti (1930, 1937/1980) called random variables with this property *exchangeable*:

> **Definition** (de Finetti, 1930): $\{Y_i, i = 1, \ldots, n\}$ are *exchangeable* if the distributions of $(Y_1, \ldots, Y_n)$ and $(Y_{\pi(1)}, \ldots, Y_{\pi(n)})$ are the same for all permutations $(\pi(1), \ldots, \pi(n))$.

NB Exchangeability and IID are not the same: exchangeable Y_i do have identical marginal distributions but are not independent. For example, if you were expecting *a priori* about 15% 1's, say (that's the 30-day death rate for AMI in England with average-quality care), the knowledge that in the first 50 outcomes 20 of them were deaths would certainly change your prediction of the 51st. In other words, for you $P_B(Y_{51} | \sum_{i=1}^{50} Y_i = 20) \neq P_B(Y_{51})$ (we will see a bit later that the Y_i only become independent *conditional on* the same θ that arose in the frequentist approach; Problem 1.1).

de Finetti also defined **partial** or **conditional** exchangeability (e.g., Draper et al., 1993): if, for instance, the gender X of the AMI patients is available, and there is evidence from the medical literature that 1's tended to be noticeably more likely for men than women, then you would probably want to assume *conditional* exchangeability of the Y_i given X_i, meaning that the male and female 1's and 0's, viewed as separate collections of random variables, are each unconditionally exchangeable.

de Finetti's representation theorem for 1's and 0's.

The judgment of exchangeability still seems to leave the joint distribution of the Y_i quite imprecisely specified. After defining the concept of exchangeability, however, de Finetti went on to prove a remarkable result: if you are willing to regard the $\{Y_i, i = 1, \ldots, n\}$ as part of an *infinite*[11] exchangeable sequence of 1's and 0's (meaning that every finite subsequence is exchangeable), then you can

express your joint distribution in a particularly simple way (e.g., de Finetti, 1975; Bernardo and Smith, 1994):

Theorem (de Finetti, 1930): If $Y_1, Y_2, \ldots$ is an infinitely exchangeable sequence of 0–1 random quantities with probability measure P, there exists a distribution function $Q(\theta)$ such that the joint distribution $p(y_1, \ldots, y_n)$ for $Y_1, \ldots, Y_n$ is of the form

$$\begin{aligned} p(y_1, \ldots, y_n) &= \int_0^1 \prod_{i=1}^n \theta^{y_i}(1-\theta)^{1-y_i}\, dQ(\theta), \\ \text{where} \quad Q(\theta) &= \lim_{n\to\infty} P(\frac{1}{n}\sum_{i=1}^n Y_i \le \theta) \qquad (1.7) \\ \text{and} \quad \theta &\stackrel{P}{=} \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^n Y_i. \end{aligned}$$

Leaving aside for a moment the interpretation of θ, the distribution function Q will generally be well-behaved enough to have a density: $dQ(\theta) = p(\theta)\, d\theta$. In this case de Finetti's Theorem says

$$p(y_1, \ldots, y_n) = \int_0^1 \prod_{i=1}^n \theta^{y_i}(1-\theta)^{1-y_i}\, p(\theta)\, d\theta. \qquad (1.8)$$

Now by the law of total probability and the definition of conditional probability,

$$\begin{aligned} p(y_1, \ldots, y_n) &= \int_0^1 p(y_1, \ldots, y_n, \theta)\, d\theta \\ &= \int_0^1 p(y_1, \ldots, y_n|\theta)\, p(\theta)\, d\theta, \qquad (1.9) \end{aligned}$$

and (1.8) and (1.9) together imply that

$$p(y_1, \ldots, y_n|\theta) = \prod_{i=1}^n \theta^{y_i}(1-\theta)^{1-y_i}. \qquad (1.10)$$

But the right side of (1.10) is just the sampling distribution of n Bernoulli random variables with common success probability θ.

Thus, according to de Finetti's Theorem, under exchangeability it is *as if* (a) there were a random quantity called θ, interpretable as the *limiting relative frequency of 1's*, (b) conditional on this θ the Y_i are IID B(θ), and (c) θ itself has a distribution with density $p(\theta)$.

In other words, a Bayesian whose uncertainty about dichotomous Y_i is exchangeable may as well use[12] the simple model

$$\begin{aligned} \theta &\sim p(\theta) \\ (Y_i|\theta) &\overset{\text{IID}}{\sim} \text{B}(\theta),\ i = 1, \ldots, n. \end{aligned} \tag{1.11}$$

This is an example of the simplest kind of **hierarchical model (HM)**[13]: a model at the top level for the underlying death rate θ, and then a model below that for the 0–1 mortality indicators Y_i conditional on θ.

1.3 Prior, posterior, and predictive distributions

Notice that to make sense of de Finetti's Theorem *you have to treat θ as a random variable*, even though logically it is a fixed unknown constant. This is the main conceptual difference between the Bayesian and frequentist approaches: as a frequentist the random variables are supposed to capture relevant features of the process of sampling from a population, whereas in the Bayesian approach you use the machinery of random variables to express your uncertainty about unknown quantities.

$\underline{\mathbf{Q}_1}$: What is the real-world meaning of $p(\theta)$ in (1.11)?

$\underline{\mathbf{A}_1}$: $p(\theta)$ does not involve $Y = (Y_1, \ldots, Y_n)$, and probability is all about uncertainty quantification for Bayesians, so $p(\theta)$ must represent your uncertainty about θ before the data set Y arrives, which is why everybody calls it your **prior distribution**[14] for θ. I will address how you might go about specifying this distribution below.

NB You don't need to literally think of θ as having been sampled from $p(\theta)$; the assumption $\theta \sim p(\theta)$ is just a way of quantifying what (if anything) was known about θ before Y is observed.

$\underline{\mathbf{Q}_2}$: If $p(\theta)$ represents your uncertainty about θ before the data arrive, what represents this uncertainty *after* Y has been observed?

$\underline{\mathbf{A}_2}$: It has to be $p(\theta|Y)$, the conditional distribution for θ given how Y came out. It is natural to call this the **posterior distribution** for θ given Y.

$\underline{\mathbf{Q}_3}$: How do you get from $p(\theta)$ to $p(\theta|Y)$—in other words, how do you update your uncertainty about the unknown θ in light of the data?

$\underline{\mathbf{A}_3}$: Use the definition of conditional probability on $p(\theta|Y)$,

$$p(\theta|Y) = \frac{p(\theta, Y)}{p(Y)}, \tag{1.12}$$

and then use the definition again to force $p(\theta)$ to appear on the right-hand side:

$$\frac{p(\theta, Y)}{p(Y)} = \frac{p(\theta)\, p(Y|\theta)}{p(Y)}. \tag{1.13}$$

The result is

Theorem (Bayes, 1763), for continuous quantities θ (the unknown) and Y (the data):

$$p(\theta|Y) = \frac{p(\theta)\, p(Y|\theta)}{p(Y)}. \tag{1.14}$$

It may seem, from how easy it is to arrive at this result, that the Rev. Bayes[15] didn't have to work very hard to achieve his immortality, but he actually did quite a bit more: he helped to put conditional probability on a sound footing for the first time, and he encouraged application of the theorem to social and medical problems, by viewing what I have here called θ and Y as examples of causes and effects, respectively—in other words, he suggested how to pass from the easier problem of predicting the likely effects of known causes to the more difficult task of inferring the causes of observed effects.

To put (1.14) into practice, some interpreting is required. As a Bayesian I want to condition on things I know and believe, in using probability to express my uncertainty: remember the $\mathcal{B}$ in equation (1.3), which I have somewhat lazily been notationally suppressing. After the data vector Y is observed, I know it—it becomes part of my $\mathcal{B}$—and so I should *condition on the data* in applying (1.14). Thus I am thinking of the left side of (1.14) as a function of θ for fixed Y, so that must also be true of the right side. In other words, (a) $p(Y)$ is just a constant—in fact, you can think of it as the **normalizing constant**, put into the equation to make the right side of (1.14) integrate to 1; and (b) $p(Y|\theta)$ may look like the usual frequentist sampling distribution for Y given θ (Bernoulli, in this case), but to use (1.14) I have to think of $p(Y|\theta)$ as a function of θ for fixed Y. When thought of this way—you could denote it $l(\theta|Y) = p(Y|\theta)$—Fisher (1922) called it the **likelihood function.**

NB The roles of θ and Y are completely reversed in the Bayesian approach to inference when compared with the frequentist approach: with my frequentist hat on I regard θ as a fixed (unknown) constant and Y as a random variable, and everything focuses on

imagining what would happen as Y changes randomly from sample to sample; but with my Bayesian hat on I am thinking of Y as a fixed (known) constant and θ as a random variable, and everything comes down to assessing my uncertainty about θ after conditioning on the one and only one Y I'm ever going to see.

From (1.14), Bayes' Theorem can evidently be interpreted as follows:

$$\begin{aligned} p(\theta|Y) &= c \cdot p(\theta) \cdot l(\theta|Y) \qquad (1.15) \\ \text{posterior} &= \begin{pmatrix} \text{normalizing} \\ \text{constant} \end{pmatrix} \cdot \text{prior} \cdot \text{likelihood.} \end{aligned}$$

You can also readily construct **predictive distributions** for the Y_i before they are observed, or for future Y_i once some of them are known. For example, the *posterior predictive distribution* for $(Y_{m+1}, \ldots, Y_n)$ given $(Y_1, \ldots, Y_m)$—that is, $p(y_{m+1}, \ldots, y_n|y_1, \ldots, y_m)$—is, by a trick similar to that in equation (1.9), just

$$\begin{aligned} &\int_0^1 p(y_{m+1}, \ldots, y_n|\theta, y_1, \ldots, y_m)\, p(\theta|y_1, \ldots, y_m)\, d\theta \\ &\qquad = \int_0^1 p(y_{m+1}, \ldots, y_n|\theta)\, p(\theta|y_1, \ldots, y_m)\, d\theta \qquad (1.16) \\ &\qquad = \int_0^1 \prod_{i=m+1}^{n} \theta^{y_i}(1-\theta)^{1-y_i}\, p(\theta|y_1, \ldots, y_m)\, d\theta\ . \end{aligned}$$

Notice an important simplification here in going from $p(y_{m+1}, \ldots, y_n|\theta, y_1, \ldots, y_m)$ to $p(y_{m+1}, \ldots, y_n|\theta)$: conditional on θ the Y_i are independent—in other words, if you know θ the individual values of $Y_1, \ldots, Y_m$ will not help you to predict $Y_{m+1}, \ldots, Y_n$. Two nice things follow from this: (a) $p(y_{m+1}, \ldots, y_n|\theta, y_1, \ldots, y_m)$ reduces to $p(y_{m+1}, \ldots, y_n|\theta)$, and then (b) since the Y_i are conditionally independent given θ, $p(y_{m+1}, \ldots, y_n|\theta)$ reduces to $\prod_{i=m+1}^{n} p(y_i|\theta)$.The result—for example, the middle line of (1.16)—is intuitively reasonable: you are trying to construct your predictive distribution for a bunch of new Ys, and it would sure help to know θ in doing so, but θ's value is not certain. So take a *weighted average*, or *mixture*, of conditional predictive distributions given θ, weighted by your best current information about θ, namely the posterior for θ given the Ys you have already seen.

This also brings up a key difference between a parameter like θ on the one hand and the Y_i, before you have observed any data,

on the other: parameters are inherently *unobservable*. This makes it harder to evaluate the quality of your uncertainty assessments about θ than to do so about the observable Y_i. Once you have the posterior for θ given Y, $p(\theta|Y)$, there is no direct way to check its quality as an uncertainty assessment, because θ is (and presumably always will remain) unknown, whereas once you have a predictive distribution $p(y_{m+1}|y_1, \ldots, y_m)$ for an observable like Y_{m+1}, you can directly check its quality by comparing the actual Y_{m+1} with your predictive distribution for it.

1.4 Inference and prediction

The de Finetti approach to modeling emphasizes the **prediction** of observables as a valuable adjunct to **inference** about unobservable parameters, for at least two reasons:

- Key scientific questions are often predictive in nature: for instance, rather than asking "Is drug A better than B (on average) for lowering blood pressure?" (inference), the ultimate question is "How much more will drug A lower *this patient*'s blood pressure than drug B?" (prediction); and
- Good *diagnostic checking* is predictive: As noted above, an inference about an unobservable parameter can never be directly verified, but often you can reasonably conclude that inferences about the parameters of a model which produces poor predictions of observables are also suspect. This will serve as the basis of the **model diagnostics** in Chapter 3.

With the predictive approach parameters diminish in importance, especially those that have no physical meaning—from the Bayesian viewpoint (e.g., Lindley, 1972) such parameters (unlike θ above) can be regarded as just *place-holders for a particular kind of uncertainty on your way to making good predictions*. It is arguable (e.g., Draper, 1995a) that the discipline of statistics, and particularly its applications in the social sciences, would be improved by a greater emphasis on predictive feedback. When was the last time you saw a statistical application, outside of (say) weather-forecasting, in which the investigators made testable predictions based on their inferential conclusions and verified them with new data?

This is not to say that parametric thinking should be abolished. As the calculation in equation (1.16) emphasized, parameters play

an important simplifying role in forming modeling judgments: the single strongest simplifier of a joint distribution is *independence* of its components, and whereas (for instance) in the mortality example the Y_i are not themselves independent, they become so conditional on θ.

1.5 Coherence and calibration

de Finetti's Theorem for 0–1 outcomes says informally that if you are trying to make **coherent**[16] (internally consistent) probability assessments about a series of 1's and 0's that you judge exchangeable, you may as well behave like a frequentist—IID B(θ)—with a prior distribution $p(\theta)$. But where does this prior come from? (**NB** Coherence doesn't help in answering this question—it turns out that *any* prior $p(\theta)$ could be part of *somebody*'s coherent probability judgments.)

Some people regard the need to answer this question in the Bayesian approach as a drawback, but it seems to me to be a positive aspect[17], as follows. From Bayes' Theorem the prior is supposed to be a summary of what you know (and don't know) about θ before the Y_i start to arrive: from previous datasets of which you are aware, from the relevant literature, from expert opinion, and so on—from all "good" sources, if any exist. Such information is almost always present, and should presumably be used when available. The issue is how to do so "well."

> The goal is evidently to choose a prior that you will retrospectively be proud of, in the sense that your predictive distributions for the observables (a) are well-centered near the actual values and (b) have uncertainty bands that correspond well to the realized discrepancies between actual and predicted values. This is a form of **calibration** of your probability judgments.

There is no guaranteed way to do this, just as there is no guaranteed way to arrive at a "good" frequentist model (see "Where does the likelihood come from?" in Section 1.8).

Choosing a "good" prior. Some general comments on arriving at a "good" prior:

- There is a growing literature on methodology for **elicitation** of prior information (e.g., Kadane et al., 1980; Craig et al., 1997;

Kadane and Wolfson, 1997; O'Hagan, 1997), which brings together ideas from statistics and perceptual psychology. To take just one example from this literature, people turn out to be better at estimating percentiles of a distribution than they are at estimating standard deviations, a fact that has direct consequences for how you should ask experts about variability.

- Bayes' Theorem on the log scale says (apart from the normalizing constant) that

$$\log(\text{posterior}) = \log(\text{prior}) + \log(\text{likelihood}); \tag{1.17}$$

in other words, (posterior information) = (prior information) + (data information). This means that close attention should be paid to the information content of the prior, for instance by density-normalizing the likelihood and plotting it on the same scale as the prior. It is possible for small n for the *prior to swamp the data*, and in general you should not let this happen without a good reason for doing so. Comfort can also be taken from the other side of this coin: with large n (in most situations, at least) (1.17) implies that the *data swamp the prior*, and prior specification errors become less important.

- When you notice you are quite uncertain about how to specify the prior, you can try **sensitivity** or **(pre-posterior) analysis**: exploring the mapping from prior to posterior, before the data are gathered, by (a) generating some possible values for the observables, (b) writing down several plausible forms for the prior, and (c) carrying these forward to posterior distributions. If the resulting distributions are similar ("all reasonable roads lead to Rome"), you have uncovered a useful form of stability in your results; if not you can try to capture the prior uncertainty hierarchically, by, for instance, adding another layer to models like (1.11) above (Problem 7.1).

- Calibration can be estimated by a form of *cross-validation:* with a given prior you can (a) repeatedly divide the data at random into modeling and validation subsets, (b) update to posterior predictive distributions based on the modeling data, and (c) compare these distributions with the actual values in the validation data. Chapter 3 illustrates some examples of this idea, which I will call **predictive validation** in what follows.

Note that calibration is inherently frequentist in spirit—it is

based on questions like "What percentage of the time do your 90% central predictive intervals include the actual value?"). This leads to a useful synthesis of Bayesian and frequentist thinking:

> Coherence keeps you internally honest; calibration keeps you in good contact with the world.

Bayes + frequentist, not Bayes vs. frequentist. People often talk about the so-called Bayesian-frequentist controversy as if it is necessary to choose sides, in a confrontation in which one approach must be right and the other wrong. There is a kind of empirical theorem that shows this attitude must be wrong: intelligent people have been arguing about this topic for almost 250 years, at least since the publication of Bayes (1763), and if the two sides were metaphorical boxers it is clear from current statistical theory and practice that both boxers are still standing in the ring after all of the punching. The implication I draw from this is that everyone should seek a personal synthesis of the best features of both the Bayesian and frequentist ways of looking at the world.

I find in my own applied work, for instance, that it is useful to reason in a Bayesian way when formulating my inferences and predictions, and to reason in a frequentist way when evaluating their quality, through calibration-style comparisons between predictive distributions for observables and the actual observables themselves. Others (e.g., Box 1980, Rubin 1984) have offered similar views; for a more skeptical position see Freedman (1995). After you have gained experience with the methods in this book, you may reach different conclusions—if so I would be interested to hear them.

1.6 Conjugate analysis

Example: Prior specification in the mortality data. Let's say (a) you know that the 30-day AMI mortality rate given average care and average sickness at admission in England is about 15% (which is in fact about right), (b) you know little about care or patient sickness at the RUH, but (c) you would be somewhat surprised (on Central Limit Theorem grounds) if the "underlying rate" at the RUH were much less than 5% or more than 30% (note the asymmetry). To quantify these judgments you seek a flexible family of densities on (0,1), one of whose members has mean 0.15 and (say) 95% central interval (0.05,0.30).

A convenient family for this purpose is the **beta** distributions, $\mathrm{Be}(\theta|\alpha,\beta) = c\,\theta^{\alpha-1}(1-\theta)^{\beta-1}$, for two reasons:

- This family exhibits a wide variety of distributional shapes (e.g., Johnson and Kotz, 1970); and
- The likelihood in this problem comes from the Bernoulli/binomial sampling distribution for the Y_i, $p(y_1,\ldots,y_n|\theta) = l(\theta|y) = c\,\theta^S(1-\theta)^{n-S}$, where $S = \sum_{i=1}^n y_i$. Thus with this choice of prior, the likelihood and prior (and thus the posterior) have the same distributional form, $\theta^q(1-\theta)^r$, which makes life computationally much easier. For this reason the collection of beta prior distributions is said to be **conjugate** to the Bernoulli/binomial likelihood[18].

Conjugate analysis—finding conjugate priors for standard likelihoods and restricting attention to them on tractability grounds—is one of only two fairly general methods for getting closed-form answers in the Bayesian approach; the other is **asymptotic analysis** (e.g., Bernardo and Smith, 1994), about which I won't have much to say here. The idea in the next few sections is to see how far conjugate analysis can take us and then to switch over to a more general approach to computation, **Markov Chain Monte Carlo (MCMC)**, in Chapter 2.

In the mortality example, trial and error shows $\alpha = 4.5$ and $\beta = 25.5$ produce approximately the desired mean and central interval—this distribution has mode 0.125 and standard deviation (SD) 0.064. α and β are called **hyperparameters** since they are parameters of the prior distribution for the parameter θ of central interest. With $(\alpha_0,\beta_0) = (4.5, 25.5)$, written hierarchically the model is

$$\begin{array}{rcll} (\alpha,\beta) & = & (\alpha_0,\beta_0) & \text{(hyperparameters)} \\ (\theta|\alpha,\beta) & \sim & \mathrm{Be}(\alpha,\beta) & \text{(prior)} \\ (Y_1,\ldots,Y_n|\theta) & \overset{\text{IID}}{\sim} & \mathrm{B}(\theta) & \text{(likelihood)} \end{array} \tag{1.18}$$

The conjugacy of the prior leads to a simple closed form for the posterior here: with y as the vector of observed $Y_i, i = 1,\ldots,n$, and S as the sum of the y_i,

$$\begin{array}{rcl} p(\theta|y,\alpha,\beta) & = & c\; p(y|\theta)\, p(\theta|\alpha,\beta) \\ & = & c\;\theta^S(1-\theta)^{n-S}\,\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ & = & c\;\theta^{(S+\alpha)-1}(1-\theta)^{(n-S+\beta)-1}; \end{array} \tag{1.19}$$

in other words, the posterior for θ is $\text{Be}(\alpha + S, \beta + n - S)$. **NB** This brings up the Bayesian version of **sufficiency**: A quantity $f(Y)$ is *sufficient* for the parameter θ—informally, $f(Y)$ is the only function of Y you need (given the model you are working with) in drawing inferences about θ—if the likelihood $l(\theta|Y)$ depends on Y only through $f(Y)$. Here S is evidently the sufficient statistic for θ with the Bernoulli/binomial likelihood.

Prior effective sample size. This gives the hyperparameters a direct interpretation in terms of *effective information content of the prior*: it is as if the data—represented by the $\text{Be}(S+1, n-S+1)$ likelihood—were worth $(S+1)+(n-S+1) \doteq n$ observations and the prior ($\text{Be}(\alpha, \beta)$) were worth $(\alpha+\beta)$ observations. This can be used to judge whether the prior is "too informative"—here it is equivalent to $(4.5 + 25.5) = 30$ binary observables with a mean of 0.15.

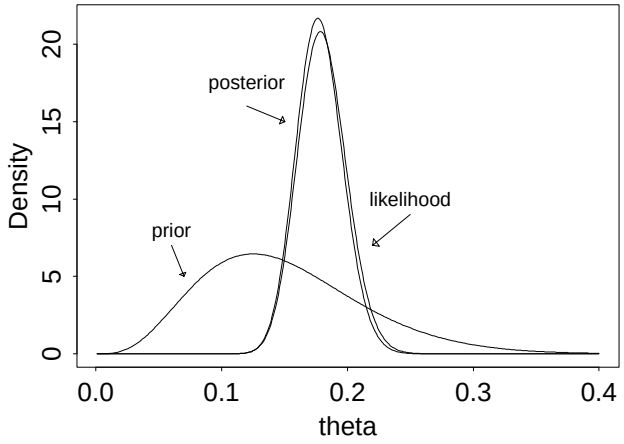

Figure 1.1. *Prior, likelihood, and posterior distributions for the RUH mortality data.*

It turns out, with conjugate models such as (1.18), that this idea can be used to make a precise connection between Bayesian and frequentist analyses of the same data set, as follows. To produce the Bayesian analysis here, it is as if you (a) create a **prior data set** with sample size $n^* = 30$, consisting of 4.5 1's and 25.5 0's (so to speak) and mean 0.15, (b) merge this prior data set with the actual data set, and (c) perform a frequentist analysis on the resulting set

of $(n^* + n)$ values. This gives a kind of literal interpretation to the idea of prior information.

Suppose the $n = 400$ observed mortality indicators consist of $S = 72$ 1's and $(n - S) = 328$ 0's. Then the prior is Be(4.5,25.5), the likelihood is Be(73,329), the posterior for θ is Be(76.5,353.5), and the three densities plotted on the same graph come out as in Figure 1.1. In this case the posterior and the likelihood nearly coincide, because the data information outweighs the prior information by 400/30, which is more than 13 to 1.

The mean of a Be(α, β) distribution is $\alpha/(\alpha + \beta)$; with this in mind the posterior mean has a clear interpretation as a weighted average of the prior mean and data mean, with weights determined by the effective sample size of the prior, $(\alpha + \beta)$, and the data sample size n:

$$\frac{\alpha+S}{\alpha+\beta+n} = \frac{\alpha+\beta}{\alpha+\beta+n} \cdot \frac{\alpha}{\alpha+\beta} + \frac{n}{\alpha+\beta+n} \cdot \frac{S}{n}$$

$$\begin{array}{ccccccccc} \text{posterior} & = & \text{prior} & \cdot & \text{prior} & + & \text{data} & \cdot & \text{data} \\ \text{mean} & & \text{weight} & & \text{mean} & & \text{weight} & & \text{mean} \end{array}$$

$$0.178 = 0.070 \cdot 0.15 + 0.93 \cdot 0.18.$$

Another way to put this is that the combining of prior and data information *shrinks* the data mean, $\bar{y} = S/n = 72/400 = 0.18$, toward the prior mean 0.15 by (in this case) a modest amount: the posterior mean is about 0.178, and the *shrinkage factor* is $30/(30+400) \doteq 0.07$. This idea of **shrinkage estimation** will come up again in Chapter 3.

1.7 Comparison with frequentist modeling

To analyze these data as a frequentist you would probably appeal to the *Central Limit Theorem*: $n = 400$ is big enough so that the sampling distribution of $\bar{Y}$ is approximately $N\left(\theta, \frac{\theta(1-\theta)}{n}\right)$, so an approximate 95% confidence interval for θ would be centered at $\hat{\theta} = \bar{y} = 0.18$, with an estimated standard error of $\sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{n}} = 0.0192$, and would run roughly from 0.142 to 0.218. By contrast the posterior for θ is also approximately Gaussian[19], with a mean of 0.178 and an SD of $\sqrt{\frac{\alpha^*\beta^*}{(\alpha^*+\beta^*)^2(\alpha^*+\beta^*+1)}} = 0.0184$, where α^* and β^* are the parameters of the beta posterior distribution; a 95% central posterior interval for θ would thus run from about 0.143 to 0.215. The two approaches give almost the same answers in this

case, a result that is typical of situations with fairly large n and relatively **diffuse** prior information (meaning that the prior SD is large relative to the normalized likelihood SD, or equivalently that the prior effective sample size is small relative to the data sample size), as in Figure 1.1.

Note, however, that the *interpretation* of the two analyses differs somewhat:

- In the frequentist approach θ is fixed but unknown and $\bar{Y}$ is random, with the analysis based on imagining what would happen if the hypothetical random sampling were repeated, and appealing to the fact that across these repetitions $(\bar{Y} - \theta) \stackrel{\cdot}{\sim} N(0, 0.0192^2)$; whereas
- In the Bayesian approach $\bar{Y}$ is fixed at its observed value and θ is treated as random, as a means of quantifying your posterior uncertainty about it: $(\theta - \bar{Y} | \bar{Y}) \stackrel{\cdot}{\sim} N(0, 0.0184^2)$.

This means among other things that, while it is not legitimate with the frequentist approach to say that $P_F(0.14 \leq \theta \leq 0.22) \doteq 0.95$, which is what many users of confidence intervals would like them to mean, the corresponding statement $P_B(0.14 \leq \theta \leq 0.22 \,|\, Y$, little or no prior info$) \doteq 0.95$ is a natural consequence of the Bayesian approach. In the case of diffuse prior information this justifies the fairly common practice of *computing inferential summaries in a frequentist way and then interpreting them Bayesianly*.

When nondiffuse prior information is available and you use it, your answer will differ from a frequentist analysis based on the same likelihood. Assuming that after the fact the likelihood is judged to have been based on an accurate reflection of the sampling realities, if your prior is retrospectively seen to have been well-calibrated you will get a better answer than with the frequentist approach; if poorly calibrated, a worse answer (Samaniego and Reneau, 1994). This may be restated schematically as

Bayesian with "bad" prior and "good" likelihood	$\leq$	frequentist with "good" likelihood	$\leq$	Bayesian with "good" prior and "good" likelihood	(1.20)

What you make of this depends on your risk-aversion: Is it better

to try to land on the right in this box, running some risk of landing on the left, or to steer a middle course? (For myself, I try to use predictive calibration (as in Section 1.5) to end up on the right. **NB** (1) I will give several examples later in which a Bayesian analysis is better even with diffuse prior information. (2) Expression (1.20) says nothing about analysts, Bayesian or frequentist, with "bad" likelihoods.)

1.8 Continuous outcomes

For continuous outcomes there is an analogue of de Finetti's Theorem that is equally central to Bayesian model-building (e.g., Bernardo and Smith, 1994):

> **Theorem** (de Finetti, 1937): If $Y_1, Y_2, \ldots$ is an infinitely exchangeable sequence of real-valued random quantities with probability measure P, there exists a probability measure Q over $\mathcal{D}$, the space of all distribution functions on the real line $\Re$, such that the joint distribution function of $Y_1, \ldots, Y_n$ has the form
>
> $$P(y_1, \ldots, y_n) = \int_{\mathcal{D}} \prod_{i=1}^{n} F(y_i)\, dQ(F), \qquad (1.21)$$
>
> where $Q(F) = \lim_{n\to\infty} P(\hat{F}_n)$ and $\hat{F}_n$ is the empirical distribution function based on $Y_1, \ldots, Y_n$.

In other words, exchangeability of real-valued observables may be taken as equivalent to the HM

$$\begin{array}{rcll} F & \sim & p(F) & \text{(prior)} \\ (Y_1, \ldots, Y_n | F) & \overset{\text{IID}}{\sim} & F & \text{(likelihood)} \end{array} \qquad (1.22)$$

for some prior distribution p on the set $\mathcal{D}$ of all possible distribution functions on $\Re$.

This prior makes the continuous form of de Finetti's Theorem considerably harder to apply: to take the elicitation task seriously is to try to specify a measure on function space (F is in effect an *infinite-dimensional* parameter). (**NB** This task is not unique to Bayesians—you may just as well ask "Where does the likelihood come from?" in frequentist analyses of observational data as to ask "Where does the prior on the parameters come from?" in Bayesian modeling.)

The field of **Bayesian nonparametrics**, which began with work by Freedman (1963), Ferguson (1973, 1974), and others, has developed in an effort to put truly rich priors on $\mathcal{D}$. This approach, however, has been stalled at an insufficiently practical stage until quite recently because of computational difficulties, but MCMC (Chapter 2) is changing that (Walker et al., 1997) at present. I will revisit this topic in Section 7.6.

Model uncertainty. Given that Bayesian nonparametrics is still basically at the pure research stage, what most Bayesians *say* they do in practice is to appeal to considerations that narrow down the field, such as an *a priori* judgment that the Y_i ought to be *symmetrically* distributed about a location parameter μ, and then try to use a plausible parametric family (the most popular is of course the Gaussian) satisfying (for instance) the symmetry restriction as a substitute for all of $\mathcal{D}$. What most analysts (Bayesian and frequentist) *actually* do in practice is to look at the data when specifying their models: for example, with data on hospital length of stay for AMI patients, you might (a) make a histogram or kernel density trace of your sample $y_1, \ldots, y_n$, (b) observe that the sample looks a lot like it follows a lognormal (LN) distribution, and (c) replace the *infinite*-dimensional elicitation problem in the first line of (1.22) by a *two*-dimensional elicitation problem on the parameters of the lognormal family. In other words, you would replace (1.22) with the vastly simpler HM

$$\begin{aligned} (\mu, \sigma^2) &\sim p(\mu, \sigma^2) && \text{(prior)} \\ (Y_1, \ldots, Y_n | \mu, \sigma^2) &\overset{\text{IID}}{\sim} LN(\mu, \sigma^2). && \text{(likelihood)} \end{aligned} \quad (1.23)$$

Now the something-for-nothing bell should be going off in your head at this point: aren't we using the data twice with this approach (once to specify the prior on $\mathcal{D}$, and once to draw inferences and make predictions given this choice of prior) and shouldn't we have to pay some price for doing so? This is the general problem of **model uncertainty** (e.g., Madigan and Raftery 1994, Draper 1995b), and it is not unique to Bayesians.

There is a real dilemma here: if you employ strategy $S^* = \{$use the data to specify the model and then pretend you knew the resulting model all along$\}$, your conclusions are likely to be miscalibrated, in the direction of underpropagation of uncertainty (in other words, your nominal 90% predictive intervals may in fact only cover the actual observables (say) 65% of the time); but not

using S^* can permit the data to surprise you in ways that would make you want to go back and revise your prior.

NB This last point is an example of what Lindley (19xx) calls **Cromwell's Rule**[20], which reminds us that it is dangerous to place prior probability 0 (or 1, for that matter; Problem 1.2) on anything, because it is then impossible to learn from any future data. For example, with any proposition A, setting $P(A) = 0$ forces $P(A \mid \text{data}) = P(A) \frac{P(\text{data}|A)}{P(\text{data})} = 0$, even if the data are highly likely under A and highly unlikely under (not A). The application of this to model uncertainty is unfortunate: in practice people generally put nonzero probability on extremely small subsets of {all possible models for a given data set}, and yet doing so without looking at the data in effect forces many things that could easily be possible (*a priori*) in the data to be impossible in your posterior analysis.

I will suggest a (partial) way out of this dilemma in Chapter 3, based on predictive validation. For the rest of the book, *faute de mieux*, I will generally either identify and work with conventional modeling choices, just to show where they lead, or use the S^*-plus-predictive-validation strategy.

Table 1.1. *NB10 frequency distribution.*

Value	375	392	393	397	398	399	400	401
Frequency	1	1	1	1	2	7	4	12
Value	402	403	404	405	406	407	408	409
Frequency	8	6	9	5	12	8	5	5
Value	410	411	412	413	415	418	423	437
Frequency	4	1	3	1	1	1	1	1

Case study 1.2: *Measurement of physical constants.* What is now called the National Institute for Standards and Technology (NIST) in Washington, DC conducts extremely high precision measurement of physical constants, such as the actual weight of so-called *check-weights* that are supposed to serve as reference standards (like the official kg). In 1962–63, for example, back when their name was the National Bureau of Standards (NBS), $n = 100$ weighings (Table 1.1) of a block of metal called NB10, which was supposed to weigh exactly 10g, were made under conditions as close to IID as possible (Freedman et al., 1998). Figure 1.2 is a

normal qqplot of the 100 measurements $y_1, \ldots, y_n$, which have a mean of $\bar{y} = 404.6$ (the units are micrograms below 10g) and an SD of $s = 6.5$.

Some natural questions that arise from the data in Table 1.1 include (a) How much does NB10 really weigh? (b) How certain are you given the data that the true weight of NB10 is less than (say) 405.25? (c) What is the underlying accuracy of the NB10 measuring process? And (d) How accurately can you predict the 101st measurement?

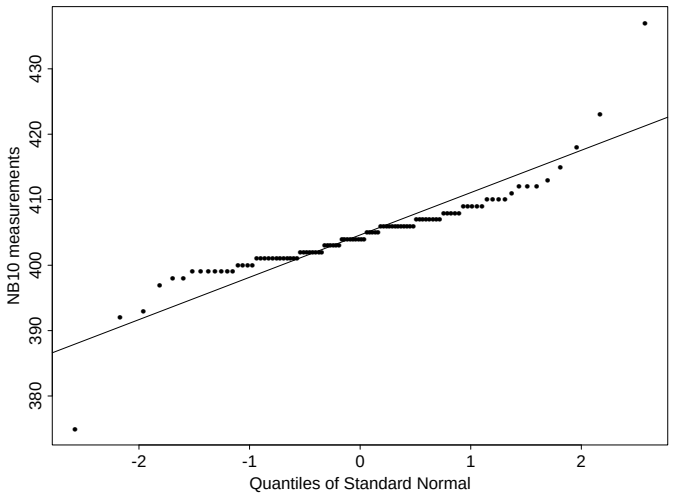

Figure 1.2. *Normal qqplot of the NB10 data.*

A simple Gaussian model. Evidently from Figure 1.2 it is plausible in answering these questions to assume symmetry of the "underlying distribution" F in de Finetti's Theorem. One conventional choice, for instance, is the *Gaussian*:

$$\begin{aligned} (\mu, \sigma^2) &\sim p(\mu, \sigma^2) \\ (Y_i \mid \mu, \sigma^2) &\overset{\text{IID}}{\sim} N(\mu, \sigma^2). \end{aligned} \tag{1.24}$$

NB People call the reciprocal of the variance σ^2 the **precision** of a distribution. In Bayesian work the precision is often the most intuitive scale on which to think about uncertainty or variability, as the results below will demonstrate.

(1.24) is our first example with more than one parameter, and we are still in the world of conjugate analysis because that's the

only computational tool I've discussed so far. Before I start in on how to specify the conjugate prior for μ and σ^2 in this model, it is helpful to look at what happens in the simpler case in which you pretend that σ^2 is known. The conjugate prior for μ (see Note 1.18) turns out (not too surprisingly, I guess) to be Gaussian; with this choice the model becomes

$$\begin{aligned} \mu &\sim N(\mu_0, \sigma_\mu^2) \\ (Y_i \mid \mu) &\overset{\text{IID}}{\sim} N(\mu, \sigma^2) \end{aligned} \tag{1.25}$$

From the conjugacy of the prior, the posterior for μ is also Gaussian, and it turns out that the posterior mean and variance have particularly simple expressions. Intuitively what is going on is this:

- The prior, considered as a data source, is Gaussian with mean μ_0, variance σ_μ^2, and precision $\frac{1}{\sigma_\mu^2}$.
- Notice from (1.19) that, in the mortality example of Section 1.6, the likelihood and posterior distributions depend on the data only through the sufficient statistic for the Bernoulli/binomial sampling distribution. In the same way you can (as people say) *reduce by sufficiency* here as well: the sufficient statistic for μ in the Gaussian model with known variance is the sample mean $\bar{Y}$. So to work out the form of the likelihood you just consider the sampling distribution of $\bar{Y}$—which is Gaussian with mean μ, variance $\frac{\sigma^2}{n}$, and precision $\frac{n}{\sigma^2}$—as a function of μ for fixed $\bar{Y}$. This distribution can be written $c_1 \exp\left[-c_2(\bar{Y}-\mu)^2\right]$, and from this you can see that $\bar{Y}$ and μ play a symmetric role in it. So if I interchange the role of μ and $\bar{Y}$, I just get a Gaussian with the same variance and precision—$\frac{\sigma^2}{n}$ and $\frac{n}{\sigma^2}$, respectively—but now it's a distribution for μ with mean $\bar{Y}$.
- In the mortality example the posterior mean was a weighted average of the prior mean and the data mean, with weights given by the prior effective sample size n^* and the data sample size n. *This turns out to be a general result with conjugate analysis*, so the same trick applies here, but what is n^* in this case? You can show that the right weights in the weighted average are given by the *precisions* of the prior and likelihood data sources:

$$E(\mu|\bar{y}) = \frac{\left(\frac{1}{\sigma_\mu^2}\right)\mu_0 + \left(\frac{n}{\sigma^2}\right)\bar{y}}{\left(\frac{1}{\sigma_\mu^2}\right) + \left(\frac{n}{\sigma^2}\right)} = \frac{\left(\frac{\sigma^2}{\sigma_\mu^2}\right)\mu_0 + n\,\bar{y}}{\left(\frac{\sigma^2}{\sigma_\mu^2}\right) + n}. \tag{1.26}$$

This also demonstrates along the way that $n^* = \frac{\sigma^2}{\sigma_\mu^2}$.

- Finally, what about the posterior variance $V(\mu|y)$? Based on what happened with the prior mean, you can guess that the posterior variance would be driven by the prior and likelihood precisions, and in fact it turns out that on the precision scale *the accuracy of the information sources is additive* (which is why Bayesians like the idea of precision so much):

$$\begin{pmatrix} \text{posterior} \\ \text{precision} \end{pmatrix} = \begin{pmatrix} \text{prior} \\ \text{precision} \end{pmatrix} + \begin{pmatrix} \text{likelihood} \\ \text{precision} \end{pmatrix}, \quad (1.27)$$

from which

$$V(\mu|y) = \frac{1}{\left(\frac{1}{\sigma_\mu^2}\right) + \left(\frac{n}{\sigma^2}\right)} = \frac{\sigma^2}{n^* + n}. \quad (1.28)$$

Some unpleasant algebra, with which I will not burden you, verifies all of the above intuition. A few points to note:

- The idea, from Section 1.6, of the prior being equivalent to a data set works again in this case: it is as if a data set with n^* observations and mean μ_0 were merged with the observed data set y and a frequentist analysis were conducted on the merged data. *This is also a general feature of conjugate analysis.*
- The concept of little or no prior information here corresponds to the prior SD σ_μ being large, or equivalently the prior precision $\frac{1}{\sigma_\mu^2}$ being small. In the limit as $\sigma_\mu \to \infty$ the prior sample size would go to 0, and you can see from (1.26) and (1.28) that the Bayesian results would coincide with the usual frequentist answers.

NB I have been using the term *diffuse* to convey the idea of a prior distribution embodying little or no information about the parameter in question. Many other Bayesians talk about **non-informative** priors in this situation, but I don't like this terminology, because every choice of prior (diffuse or not) conveys information, namely your choice—which needs to be defended in each case—for the appropriate effective prior sample size. I will stick with *diffuse* in what follows.

Bayesian inference with multivariate θ. Returning now to (1.24) with σ^2 unknown, this model has a ($p = 2$)-dimensional parameter, $\theta = (\mu, \sigma^2)$. When $p > 1$ you can still use Bayes' Theorem

directly to obtain the **joint posterior distribution**,

$$\begin{aligned} p(\theta \mid y) &= c\, p(\theta)\, l(\theta \mid y) = p(\mu, \sigma^2 \mid y) \\ &= c\, p(\mu, \sigma^2)\, l(\mu, \sigma^2 \mid y), \end{aligned} \tag{1.29}$$

where $y = (y_1, \ldots, y_n)$, although making this calculation directly requires a p-dimensional integration to evaluate c—for example, in this case

$$\begin{aligned} c &= [p(y)]^{-1} = \left(\iint p(\mu, \sigma^2, y)\, d\mu\, d\sigma^2 \right)^{-1} \\ &= \left(\iint p(\mu, \sigma^2)\, l(\mu, \sigma^2 \mid y)\, d\mu\, d\sigma^2 \right)^{-1}. \end{aligned} \tag{1.30}$$

Usually, however, you will be more interested in the **marginal posterior distributions**, in this case $p(\mu \mid y)$ and $p(\sigma^2 \mid y)$. Obtaining these requires p integrations, each of dimension $(p-1)$, a process that people refer to as *marginalization* or *integrating out the nuisance parameters*. For example,

$$p(\mu \mid y) = \int p(\mu, \sigma^2 \mid y)\, d\sigma^2 \ . \tag{1.31}$$

Predictive distributions also involve a p-dimensional integration: for example, with $y = (y_1, \ldots, y_n)$,

$$\begin{aligned} p(y_{n+1} \mid y) &= \iint p(y_{n+1}, \mu, \sigma^2 \mid y)\, d\mu\, d\sigma^2 \\ &= \iint p(y_{n+1} \mid \mu, \sigma^2)\, p(\mu, \sigma^2 \mid y)\, d\mu\, d\sigma^2. \end{aligned} \tag{1.32}$$

And, finally, if you are interested in a *function of the parameters*, you have some more hard integrations ahead of you. For instance, suppose you wanted the posterior distribution for the *coefficient of variation* $\lambda = g_1(\mu, \sigma^2) = \frac{\mu}{\sqrt{\sigma^2}}$ in model (1.24). Then one fairly direct way to get this posterior (e.g., Bernardo and Smith, 1994) is to (a) introduce a second function of the parameters, say $\eta = g_2(\mu, \sigma^2)$, such that the mapping $f = (g_1, g_2)$ from (μ, σ^2) to (λ, η) is invertible; (b) compute the joint posterior for (λ, η) through the usual change-of-variables formula

$$p(\lambda, \eta | y) = p_{\mu,\sigma^2}\left[f^{-1}(\lambda, \eta) | y\right] \ \left| J_{f^{-1}}(\lambda, \eta) \right| , \tag{1.33}$$

where $p_{\mu,\sigma^2}(\cdot, \cdot | y)$ is the joint posterior for μ and σ^2 and $\left| J_{f^{-1}} \right|$ is the determinant of the Jacobian of the inverse transformation;

and (c) marginalize in λ by integrating out η in $p(\lambda, \eta|y)$, in a manner analogous to (1.31). (Here, for instance, $\eta = g_2(\mu, \sigma^2) = \sqrt{\sigma^2}$ would create an invertible f, with inverse defined by ($\mu = \lambda\eta, \sigma^2 = \eta^2$); the Jacobian determinant comes out $2\lambda\eta$ and (1.33) becomes $p(\lambda, \eta|y) = 2\lambda\eta\, p_{\mu,\sigma^2}(\lambda\eta, \eta^2|y)$.) This process involves two integrations, one to get the normalizing constant that defines (1.33) and one to get rid of η.

You can see that when p is a lot bigger than 2 all these integrals may create severe computational problems—this has been the big stumbling block for applied Bayesian work for a long time.

More than 200 years ago Laplace (1774)—perhaps the second applied Bayesian in history (after Bayes himself)—developed, as one avenue of solution to this problem, what people now call **Laplace approximations** to high-dimensional integrals of the type arising in Bayesian calculations (see, e.g., Tierney and Kadane, 1986). I will cover Laplace approximations only briefly in this book, in Chapter 8; Chapter 2 details how MCMC may be used as an alternative solution to the integration problem.

The full Gaussian case. The conjugate prior for (μ, σ^2) in the model (1.24) (e.g., Gelman et al., 1995) turns out to be most simply described hierarchically:

$$\begin{aligned} \sigma^2 &\sim SI\text{-}\chi^2(\nu_0, \sigma_0^2) \\ (\mu \,|\, \sigma^2) &\sim N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right) \end{aligned} \tag{1.34}$$

Here saying that $\sigma^2 \sim$ SI-$\chi^2(\nu_0, \sigma_0^2)$, where SI stands for *scaled inverse*, amounts to saying that $\tau^2 \equiv \frac{1}{\sigma^2}$ follows a scaled χ^2 distribution with parameters ν_0 and σ_0^2 (see Appendix 1 for details). The scaling is chosen so that σ_0^2 can be interpreted as a *prior estimate* of σ^2, with ν_0 the *prior effective sample size* of this estimate (in other words, as in the beta-Bernoulli/binomial model of Section 1.6, think of a prior data set with ν_0 observations and sample variance σ_0^2). The parameters μ_0 and κ_0 in the second level of the prior model (1.34) have simple parallel interpretations to those of σ_0^2 and ν_0: μ_0 is the prior estimate of μ, and κ_0 is the prior effective sample size of this estimate.

In the Gaussian model (1.24, 1.34), which I will abbreviate $\mathcal{G}$, the integrations may be done analytically (e.g., Gelman et al., 1995), yielding

$$\begin{aligned}
(\sigma^2 \mid y, \mathcal{G}) &\sim SI\text{-}\chi^2(\nu_n, \sigma_n^2) \\
(\mu \mid y, \mathcal{G}) &\sim t_{\nu_n}\left(\mu_n, \frac{\sigma_n^2}{\kappa_n}\right), \quad \text{where} \\
\nu_n &= \nu_0 + n, \qquad \kappa_n = \kappa_0 + n, \\
\nu_n \sigma_n^2 &= \nu_0 \sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_n}(\bar{y} - \mu_0)^2, \\
\mu_n &= \frac{\kappa_0}{\kappa_n}\mu_0 + \frac{n}{\kappa_n}\bar{y}.
\end{aligned} \tag{1.35}$$

Here $\bar{y} = \frac{1}{n}\sum_{i=1}^n$ and $s^2 = \frac{1}{n-1}\sum_{i=1}^n (y_i - \bar{y})^2$ are the usual sample mean and variance of y, and $t_\nu(\mu, \sigma^2)$ is a scaled version of the usual t_ν distribution (Appendix 1): $W \sim t_\nu(\mu, \sigma^2)$ just means that $\frac{W-\mu}{\sigma} \sim t_\nu$. Once again, from the conjugacy, the posterior mean for μ in (1.35) is a weighted average of the prior mean and data mean, with weights determined by the effective prior sample size and the data sample size.

NB10 Gaussian results. (1.35) may be used to answer the four questions listed below Table 1.1, as follows.

Question (a): I don't know anything *a priori* about what NB10 is supposed to weigh (down to the nearest microgram) or about the accuracy of the NBS's measurement process, so I want to use a diffuse prior for μ and σ^2. Considering the meaning of the hyperparameters, to provide little prior information I want to choose both ν_0 and κ_0 close to 0. Making them exactly 0 would produce an **improper** prior distribution (which doesn't integrate to 1), but choosing positive values as close to 0 as you like yields a proper and highly diffuse prior.

You can see from (1.35) that the result for large n is then

$$(\mu \mid y, \mathcal{G}) \dot{\sim} t_n\left(\bar{y}, \frac{(n-1)s^2}{n^2}\right) \doteq N\left(\bar{y}, \frac{s^2}{n}\right); \tag{1.36}$$

in other words, with diffuse prior information (and as with the Bernoulli model in Section 1.6) the 95% central Bayesian interval virtually coincides with the usual frequentist 95% confidence interval $\bar{y} \pm t_{n-1}^{.975}\frac{s}{\sqrt{n}} = 404.6 \pm 1.98 \cdot 0.647 = (403.3, 405.9)$. Thus both {frequentists who assume $\mathcal{G}$} and {Bayesians who assume $\mathcal{G}$ with a diffuse prior} conclude that NB10 weighs about 404.6μg below 10g, give or take about 0.65μg.

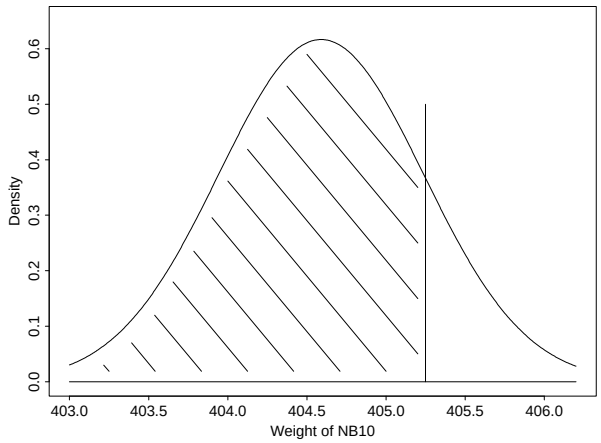

Figure 1.3. *Posterior distribution for μ in the NB10 example, with the $\{\mu < 405.25\}$ region shaded.*

Question (b). If interest focuses on whether NB10 weighs less than some value like 405.25, when reasoning in a Bayesian way you can answer this question directly: the posterior distribution for μ is shown in Figure 1.3, and $P_B(\mu < 405.25 \mid y, \mathcal{G}, \text{diffuse prior}) \doteq 0.85$. In other words, with these assumptions there are pretty good betting odds—about 5.5 to 1—in favor of the proposition that $\mu < 405.25$.

With your frequentist hat on, $P_F(\mu < 405.25)$ is undefined; about the best you can do is to test H_0: $\mu < 405.25$, for which the p-value would (approximately) be $p = P_{F,\mu=405.25}(\bar{y} > 405.59) = 1 - 0.85 = 0.15$. This would constitute "insufficient evidence to reject H_0 at the usual significance levels," leaving an inferential impression that contrasts with the reasonably clear Bayesian betting odds. **NB** (1) The significance test tries to answer a different question: in Bayesian language it looks at $P(\bar{y} \mid \mu)$ instead of $P(\mu \mid \bar{y})$. (2) You can see that as with confidence intervals, when a diffuse prior seems appropriate, there is a direct relationship—at least with one-sided tests[21]—between frequentist and Bayesian results: for testing H_0: $\mu < c$, the p-value is just $p = 1 - P_B(\mu < c \mid y, \text{diffuse prior})$. Thus there is a certain justification in one-sided testing problems for the conclusion, which people sometimes wish to draw, that the p value is the probability that the null hypothesis is false.

Question (c). The conjugacy of (1.34) means that the assumption of a scaled inverse χ^2 prior for σ^2, with hyperparameters ν_0 and σ_0^2, also produces a $SI\text{-}\chi^2$ posterior for σ^2, with the following parameters (see, e.g., Gelman et al., 1995):

$$\begin{aligned} (\sigma^2|y,\mathcal{G}) &\sim SI\text{-}\chi^2(\nu_n,\sigma_n^2), \quad \text{where} \\ \nu_n &= \nu_0 + n \qquad\qquad \text{and} \\ \sigma_n^2 &= \frac{1}{\nu_n}\left[\nu_0\sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_n}(\bar{y}-\mu_0)^2\right]. \end{aligned} \tag{1.37}$$

The form of σ_n^2, which acts (for large ν_n, at least) like a posterior estimate of σ^2, is interesting: the first two terms in σ_n^2 are (almost) a weighted average of the prior and sample estimates σ_0^2 and s^2 of σ^2, and there is also a contribution arising from the discrepancy, if any, between the sample mean $\bar{y}$ and the prior mean μ_0.

As in the answer to question (a), a diffuse prior would correspond to choosing ν_0 and κ_0 close to 0, which would produce the result

$$(\sigma^2|y,\mathcal{G},\text{diffuse prior, large } n) \dot{\sim} SI\text{-}\chi^2(n,\frac{n-1}{n}s^2). \tag{1.38}$$

Now you can also show (Problem 1.8) that (1.38) is equivalent to saying that the posterior distribution for the precision $\frac{1}{\sigma^2}$ is gamma with parameters $\frac{n}{2}$ and $\frac{n-1}{2}s^2$, which I will denote $\Gamma(\frac{n}{2},\frac{n-1}{2}s^2)$. This means that the posterior for $\frac{(n-1)\,s^2}{\sigma^2}$ is $\Gamma(\frac{n}{2},\frac{1}{2})$, which is another way of writing the χ^2 distribution with $(n-1)$ degrees of freedom. But this is just a Bayesian interpretation of the usual frequentist inference for σ^2 in the Gaussian model with both μ and σ^2 unknown (e.g., Snedecor and Cochran, 1980): the sampling distribution of $\frac{(n-1)\,s^2}{\sigma^2}$ in this model, viewing σ^2 as fixed and s^2 as random, is χ^2_{n-1}. Thus, as in the answer to question (a), {Bayesians with diffuse prior information} and {frequentists} would get the same 95% (central) intervals for σ^2 and σ: the NB10 sample SD of $s = 6.5$ produces the 95% interval estimates $(32.2, 56.4)$ and $(5.68, 7.51)$ of σ^2 and σ, respectively.

An informative prior in the Gaussian model. Simply for illustration, suppose that information from other studies at the NBS before the NB10 data were collected—taking suitable account of any differences between the previous studies and the present measurement method—had suggested that σ should be around $\sigma_0 = 10$, with (say) 90% *a priori* limits of roughly ($\sigma_{\text{lo}} = 6, \sigma_{\text{hi}} = 34$), and that μ should be around $\mu_0 = 403$ with 90% prior limits of

approximately ($\mu_{\text{lo}} = 396, \mu_{\text{hi}} = 410$). To fit this information into the conjugate structure (1.34), I have to find the corresponding prior effective sample sizes ν_0 and κ_0.

Considering ν_0 first, the fact noted above that $\sigma^2 \sim SI\text{-}\chi^2(\nu_0, \sigma_0^2)$ iff $\frac{\nu_0 \sigma_0^2}{\sigma^2} \sim \chi^2_{\nu_0}$ makes me want to work with the precision instead of the SD in setting up an equation to determine ν_0, since I have a χ^2 CDF handy in S+:

$$\begin{aligned} 0.9 &= P(\sigma_{\text{lo}} < \sigma < \sigma_{\text{hi}}) \\ &= P\left(\frac{\nu_0\sigma_0^2}{\sigma_{\text{hi}}^2} < \frac{\nu_0\sigma_0^2}{\sigma^2} < \frac{\nu_0\sigma_0^2}{\sigma_{\text{lo}}^2}\right) \\ &= F_{\chi^2_{\nu_0}}\left(\frac{100\,\nu_0}{1156}\right) - F_{\chi^2_{\nu_0}}\left(\frac{100\,\nu_0}{36}\right), \end{aligned} \tag{1.39}$$

where $F_{\chi^2_{\nu_0}}$ is the $\chi^2_{\nu_0}$ CDF. Trial and error with this CDF now shows that $\nu_0 \doteq 2.5$.

In specifying κ_0 it is helpful to appeal to a fact about the distribution of μ in the conjugate prior specification (1.34). You can show (Problem 1.10) that if σ^2 is $SI\text{-}\chi^2$ and $(\mu|\sigma^2)$ is Gaussian then the marginal distribution of μ is scaled t:

$$\text{(1.34) implies that} \quad \mu \sim t_{\nu_0}\left(\mu_0, \frac{\sigma_0^2}{\kappa_0}\right). \tag{1.40}$$

From the CDF of the standard t distribution with $\nu_0 = 2.5$ degrees of freedom,

$$\begin{aligned} P\left(\left|\frac{\mu - \mu_0}{\sqrt{\sigma_0^2/\kappa_0}}\right| \le 2.56\right) &= 0.9, \quad \text{yielding} \\ \kappa_0 = \frac{2.56^2\,\sigma_0^2}{(\mu_{\text{hi}} - \mu_0)^2} &\doteq 13. \end{aligned} \tag{1.41}$$

With these values of ν_0 and κ_0 and the NB10 data, (1.37) produces the posterior parameters $\nu_n = 102.5$ and

$$\sigma_n^2 \doteq \frac{250 + 4140.19 + 29.08}{102.5} \doteq 43.11. \tag{1.42}$$

This is a bit bigger than the sample variance $s^2 \doteq 41.82$, both because the prior estimate of σ^2 (100) is a lot bigger than s^2 and because of the modest discrepancy between the prior and data means. The prior-to-posterior analysis here is not far from the simple up-

dating rule

$$\begin{aligned} p(\sigma^2, \mathcal{G}) &= SI\text{-}\chi^2(\nu_0, \sigma_0^2), && \text{(prior)} \\ l(\sigma^2 | y, \mathcal{G}) &= SI\text{-}\chi^2(n, s^2) && \text{(likelihood)} \quad (1.43) \\ p(\sigma^2 | y, \mathcal{G}) &= SI\text{-}\chi^2(\nu_0 + n, \frac{\nu_0 \sigma_0^2 + n s^2}{\nu_0 + n}) && \text{(posterior),} \end{aligned}$$

which would have been exact in the Gaussian model if μ had been known (Problem 1.11). The three distributions in (1.43) are plotted in Figure 1.4; you can see that the prior information has tugged the posterior to the right of the data information, but not very much because the prior effective sample sizes were small.

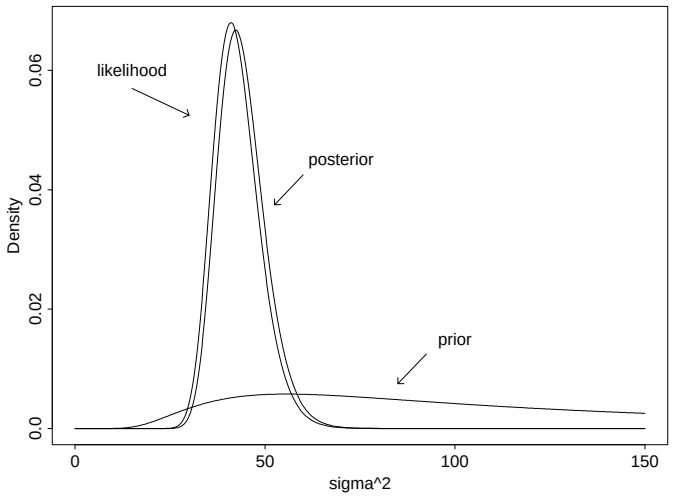

Figure 1.4. *Prior, likelihood, and posterior distributions for σ^2 with the Gaussian model* (1.24, 1.34) *and an informative prior applied to the NB10 data.*

Question (d). Analytic integration in the Gaussian model (e.g., Bernardo and Smith, 1994) yields

$$(y_{n+1} \,|\, y, \mathcal{G}) \sim t_{\nu_n}\left(\mu_n, \frac{\kappa_n + 1}{\kappa_n}\sigma_n^2\right), \qquad (1.44)$$

and for n large and ν_0 and κ_0 close to 0 this is $(y_{n+1} \,|\, y) \dot{\sim} N(\bar{y}, s^2)$ (the basis of the usual frequentist answer), yielding a 95% posterior predictive interval for y_{n+1} of (392, 418).

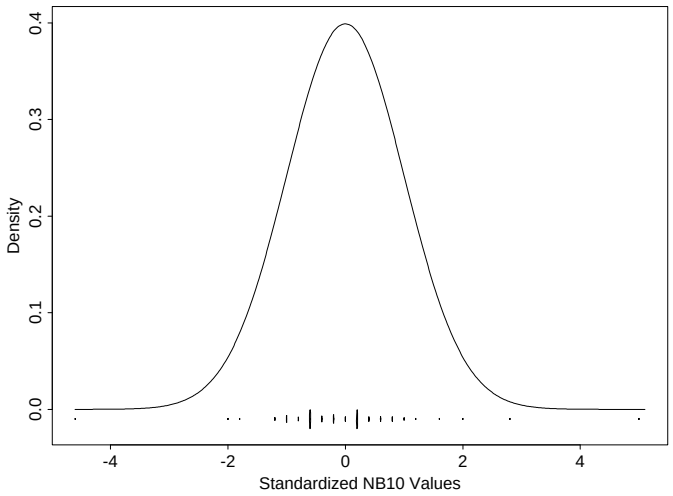

Figure 1.5. *Standardized posterior predictive distribution for y_{n+1} in the NB10 example, with a dotplot of the 100 standardized data values superimposed (symbol height is proportional to number of (tied) observations at each point).*

Model expansion. A standardized version of the predictive distribution (1.44) for the NB10 data is plotted in Figure 1.5, with the standardized data values superimposed. It is evident from this plot (and also from Figure 1.2) that the Gaussian model provides a poor fit for these data—the two most extreme points in the data set in standard units are -4.6 and 5.0. With the symmetric heavy tails indicated in these plots, in fact, the empirical CDF looks quite a bit like that of a t distribution with a rather small number of degrees of freedom ν. This suggests revising the previous model by *expanding* it: embedding the Gaussian in the t family and adding the parameter ν for tail-weight.

Unfortunately there is no standard closed-form conjugate choice for the prior on ν. A more flexible approach to computing is evidently needed: this is the subject of Chapter 2.

1.9 Additional reading

[xx finish this]

1.10 Problems

[xx this section is still quite rough]

1.1 Consider the model $\theta \sim p(\theta)$, $(Y_i|\theta) \overset{\text{IID}}{\sim} B(\theta)$ for some prior $p(\theta)$ for which the prior mean $E(\theta)$ and variance $V(\theta)$ are both nonzero. Show in this model that $P(Y_2 = 1) < P(Y_2 = 1 \,|\, Y_1 = 1)$, thereby concretely demonstrating that without conditioning on θ the Y_i are dependent. Thus IID $\rightarrow$ exchangeability but not conversely.

1.2 I said in Section 1.3 that "... with my frequentist hat on I regard θ as a fixed (unknown) constant and Y as a random variable, and everything focuses on imagining what would happen as Y changes randomly from sample to sample." This is actually the logical position for frequentists *before* the data arrive. What about *after* the data arrive—once Y is observed, in the frequentist approach is it still random, or is it now fixed? If it's still random, then is it fair to say that frequentists don't condition on the data? If it's now fixed, then both θ and Y are fixed, and where does probability come in? Discuss.

1.3 The Enzyme-Linked ImmunoSorbent Assay (*ELISA*) test was approved by many countries around the world in the mid-1980s to screen donated blood for the presence of the AIDS virus HIV. The test works by detecting antibodies—substances that the body produces when the virus is present—but, as with any screening test, in practice it makes some mistakes. *ELISA* was designed so that when a given blood sample does in fact contain a clinically meaningful concentration of HIV, the test gives a positive result (that is, *ELISA* reports that in its opinion this blood sample has HIV in it) $\alpha = 98\%$ of the time: this is referred to as *ELISA*'s *sensitivity*. Moreover, when the blood being tested is not contaminated with the virus *ELISA* will announce a negative result $\beta = 93\%$ of the time: this is *ELISA*'s *specificity*. The *prevalence* of HIV-positivity in the population of people who donate blood to blood banks is thought to be about $\pi = 1\%$.

(a) Letting $A = \{\text{person is HIV-positive}\}$ and $+ = \{\textit{ELISA} \text{ positive}\}$, express the three numerical facts above in unconditional and conditional probability terms, and use Bayes' Theorem to show that if someone donates blood and the *ELISA* test comes out negative, the probability the person is not in

fact HIV-positive given this negative result is virtually 100%, but if *ELISA* comes out positive the probability the person actually is HIV-positive is only $p = \frac{98}{791} \doteq 12\%$. Explain these results by (i) exploring symbolically and numerically how p depends on α, β, and π, and noting what it is about the given values of these three quantities that has made p so low; and (ii) identifying the two kinds of mistakes *ELISA* could make and discussing their implications from the blood bank's point of view.

(b) In practice it is possible to "tune" screening tests like *ELISA* by changing the threshold of antibodies required to announce a positive result, which will act on the 98% sensitivity and 93% specificity values mentioned above in a tug-of-war fashion: you can increase the sensitivity, for instance, but only by allowing the specificity to decrease (and vice versa). If *ELISA* were to be made available as a screening test to the general population (for instance, suppose that people were able to send a blood sample to a private lab confidentially and get back the *ELISA* diagnosis for a fee), which would it be better to increase: *ELISA*'s sensitivity or specificity? What would happen if *ELISA*, with its present α and β, were used as a public health tool in a mass screening program of all Americans, as some members of the US Congress suggested back in the 1980s? Explain.

1.4 Prove the other part of Cromwell's Rule (Section 1.8): With any proposition A, setting $P(A) = 1$ forces $P(A \mid \text{data}) = 1$ no matter how likely the data are under A and (not A).

1.5 I used to work at the University of California in Los Angeles, and I like to drink tea while I'm working. For the first few weeks after starting work I didn't have access to any facilities for making tea, so I would go down twice a day to a vending machine and pay 25 cents for a cup of brown liquid that the machine claimed was "tea." On the front of the machine there was a bright yellow label that said something like, "Maybe you'll be lucky!! Every now and then, at random, this machine will give you your quarter back, and your beverage will be free!!"

At the end of almost two months, having spent $n = 78$ quarters without getting "lucky," it occurred to me that the company that owned the machine may have just decided it was cheaper

to put the yellow label on the front than to install any randomization device inside that would actually make refunds. Let $Y_i = 1$ if I got a free cup of "tea" on occasion i and assume the Bernoulli model (1.11) with a prior that appropriately reflects the company's desire to make money (for example, do you really believe that $\theta = 0.8$ is as likely *a priori* as $\theta = 0.008$?). Use this model to discriminate between the two possible explanations $\{\theta = 0\}$ and $\{\theta > 0\}$ by calculating their posterior probabilities given the data, and explain why the frequentist p value for testing H_0: $\theta = 0$ would be completely useless in this situation[22].

1.6 Consider a univariate parameter θ and a data set $Y = (Y_1, \ldots, Y_n)$ that is IID from some sampling distribution given θ. Learning about θ from Y in the Bayesian approach involves updating from $p(\theta)$ to $p(\theta|Y)$, and it is interesting, from an experimental design point of view, to examine what may be said in general about the relationship between these two distributions before Y is observed.

(a) The first part of the double expectation theorem from introductory probability says that the prior mean $E(\theta)$ and the posterior mean $E(\theta|Y)$ are related by

$$E(\theta) = E_Y[E(\theta|Y)], \tag{1.45}$$

where the right side of (1.45) involves averaging over possible data sets Y. Explain what this implies—if you were planning on quoting the mean as a point estimate of θ—about the effect you expect the data to have on your prior point estimate. Is this intuitively reasonable? Explain.

(b) The second part of the double expectation theorem[23] says that the prior and posterior variances $V(\theta)$ and $V(\theta|Y)$, respectively, are related by

$$V(\theta) = V_Y[E(\theta|Y)] + E_Y[V(\theta|Y)]. \tag{1.46}$$

Show that this means that, averaging over possible Y, you expect to learn about θ, in the sense that you expect the posterior variance to be no larger than the prior variance. However, by creating an explicit example (prior and data set) in the beta/Bernoulli model (1.18) or the Gaussian model (1.24, 1.34), also show that it is possible for you to "know less" after you see Y than before, in that the posterior variance can

be *larger* than the prior variance. Explain in concrete terms what feature of the relationship between the prior and data information causes this to happen.

1.7 Continuing Problem 1.6, show in the simple Gaussian model (1.25) with known σ^2 that, no matter how discrepant the prior mean and the data mean are, the posterior variance will always be smaller than either the prior variance or the "data variance" (the variance of the density-normalized likelihood). What is it about (1.25) that produces this undesirable result, and what is it about the full Gaussian model (1.24, 1.34) that remedies the defect? Explain.

1.8 Relationships between gamma, inverse gamma, χ^2, inverse χ^2, scaled inverse χ^2 [xx to be finished].

1.9 ($\mathcal{N}1$) Poisson-gamma; negative binomial [xx to be finished].

1.10 Show that if σ^2 is SI-χ^2 and $(\mu|\sigma^2)$ is Gaussian then the marginal distribution of μ is scaled t [xx to be finished].

1.11 Prove (1.43) in the normal model with known mean [xx to be finished].

1.12 Consider the simple Gaussian model (1.25) and make it even simpler by taking $n = 1$: $\theta \sim N(\mu_0|\sigma_\mu^2)$, $(Y|\theta) \sim N(\mu, \sigma^2)$ for known σ^2. Before you have seen Y this is like a bivariate sampling model for (θ, Y): θ is drawn from a Gaussian, and then conditional on θ, Y is drawn from another Gaussian. This makes me think of a elliptical (why?) scatter plot for (θ, Y) and brings up the idea of *regression* as an alternative way to understand how Bayes' Theorem works in this model.

It follows from the assumptions so far that both the marginal distribution of Y and the conditional distribution of θ given Y are also Gaussian (this was Galton's original way of thinking about regression more than 100 years ago, in fact; see Stigler, 1986). Use the double expectation theorem and anything you know about regression to derive the posterior mean and variance of θ given Y.

1.13 Sequential updating: show that you get the same thing when you sequentially absorb $y_1, \ldots, y_n$ as when you simultaneously absorb them.

[I will supply more problems later, and the new problems will be more interesting and data-oriented than, e.g., 1.10 and 1.11 ... xx to be continued].

1.11 Notes

1.1 *Process* is what health care providers do on behalf of patients; *outcomes* are what happens as a result of that care. Saying that a disease has a strong process-outcome link just means that research has demonstrated for that disease that good process leads to good outcomes and bad process to bad outcomes.

1.2 In the history of ideas this branch of mathematics is relatively new—the ancient Greeks, for example, had no notion of probability. The subject seems to have come into focus fairly suddenly in about 1660, in the independent work of a variety of people including Leibniz (Germany), Pascal (France), Huygens (Holland), and Graunt (England); see Hacking (1975) and Stigler (1986).

1.3 I strongly recommend this excellent book to anyone interested in the foundations of probability and statistics. Oakes presents a devastating critique of significance testing, an interesting account of the various ways people have tried to connect probability with the real world, a comparative evaluation of the leading schools of statistical inference, and a discussion of the role of statistics in the social sciences.

1.4 For example, suppose you are trying to quantify your uncertainty about the probability θ of something happening, and you want to express the judgment that any value for θ from 0 to 1 (inclusive) is equally plausible (the *principle of insufficient reason*). OK, but what if you had asked yourself the same question about $f(\theta)$ for some monotone f, like θ^2? You cannot claim that any value for θ from 0 to 1 (inclusive) is equally plausible at the same time as you are claiming that any value for θ^2 from 0 to 1 (inclusive) is equally plausible. This was what bothered Fisher (1922) about the Bayesian need to specify a prior. However, as Lindley (19xx) points out [Dennis, please help me with a relevant reference], in the language of Note 1.17 below this is actually a feature, not a bug: even if you are pretty unsure of the value of θ, you are pretty darn sure that θ^{1000} is close to 0.

1.5 This is especially likely in sparse cells in the **equivalence grid** formed conceptually by crossing categorical versions of the predictor variables with each other.

1.6 To a Bayesian saying that $P_B(A)$ is "objective" just means that lots of people more or less agree on its value.

1.7 It has bothered me for a long time that almost nobody talks about modeling issues like this in the frequentist approach—instead, people rush directly to "Let $y_1, \ldots, y_n$ be IID," without saying anything about why random variables are appropriate with samples of convenience and data from observational studies. A few exceptions: the lovely introductory statistics book by Freedman et al. (1998), and Mallows (1998).

1.8 Of course, in practice nobody would treat this argument seriously until you compared the *observed* mortality at the RUH with its *expected* mortality given how sick its AMI patients were on admission; see, e.g., Keeler et al. (1990).

1.9 When necessary I will use the standard convention of writing random variables in upper case and the values they take on in lower-case.

1.10 I have repeated this observation several times to emphasize, for people who have so far only thought about probability from the frequentist viewpoint, that something fundamentally different is going on here with the random variables in the Bayesian approach.

1.11 Finite versions of de Finetti's Theorem are available (Diaconis and Freedman, 1980; Bernardo and Smith, 1994): call an exchangeable sequence $\{y_i, i \leq n\}$ N-**extendable** if it is the first n elements of a longer exchangeable sequence $\{y_i, i \leq N\}$. (Infinite exchangeability, as in Theorem 1.7, amounts to assuming N-extendability for all $N > n$.) Then (1.7) is a good approximation to (1.6) when $N >> n$. In practice this means that you regard the process of observing 1's and 0's to be time-homogeneous across a horizon that is considerably broader than the first n observations—in other words, we are back in effect to the frequentist difficulty of having to define a population. There is no free lunch with de Finetti's Theorem.

1.12 I used to think that de Finetti's Theorem says that if your uncertainty about the Y_i is exchangeable then you *must* express your predictive distribution (1.6) for the Y_i in the form (1.7), but (as Sander Greenland pointed out to me) in fact all the theorem says is that you *can* express it in this form. In practice, however, de Finetti's representation is so straightforward to work with that most people just move directly from the exchangeability judgment to (1.7).

1.13 This is a slightly nonstandard use of the term hierarchical model; many people think of HMs as models for situations with *data* at all levels of the hierarchy, although in such fields as **meta-analysis** (Chapter 3) even this convention is more honored in the breach than in the observance.

1.14 Since we may as well model θ as a continuous quantity between 0 and 1, $p(\theta)$ is an ordinary continuous probability density, just like any frequentist-style sampling distribution on (0,1).

1.15 Thomas Bayes (1701?–1761) was an English cleric, philosopher, and mathematician, interested in the foundations of probability and (what we would now think of as) statistics, who managed to make a place for himself in history without a single mathematical publication in his lifetime. Stigler (1986) has a lot of interesting material on what Bayes actually did and did not do. For instance, in the famous essay that he did not allow to be published until after his death, Bayes (1763) posed and solved the following problem, in present-day notation: if $\theta \sim U(0,1)$ and $(Y|\theta) \sim \text{bin}(n,\theta)$ then compute $P(a < \theta < b|Y)$ for any a and b. The main controversy concerned the universal appropriateness of his choice of a uniform prior distribution for θ in real-world problems.

1.16 The formalism of **coherence** is best understood within the context of Bayesian decision theory. Axiomatic approaches to rational decision-making date back to Ramsay (1931/1980), with von Neumann and Morgenstern (1944) and Savage (1954) also making major contributions. The ingredients of a general decision problem (e.g., Bernardo and Smith, 1994) include

- A set $\{a_i, i \in I\}$ of available **actions**, one of which you will choose;
- For each action a_i, a set $\{E_j, j \in J\}$ of **uncertain outcomes** describing what will happen if you choose action a_i;
- A set $\{c_j, j \in J\}$ of **consequences** corresponding to the outcomes $\{E_j, j \in J\}$; and
- A **preference relation** $\leq$, expressing your preferences between pairs of available actions ($a_1 \leq a_2$ means "a_1 is not preferred by you to a_2"). Define $a_1 \sim a_2$ ("a_1 and a_2 are *equivalent*" to you) iff $a_1 \leq a_2$ and $a_2 \leq a_1$.

This preference relation induces a *qualitative* ordering of the uncertain outcomes ($E \leq F$ means "E is not more likely than

F"), because if you compare two dichotomized possible actions, involving the same consequences and differing only in their uncertain outcomes, the fact that you prefer one action to another means that you must judge it more likely that if you take that action the preferred consequence will result.

Within this framework you have to make further assumptions—the **coherence** axioms—to ensure that your actions are internally consistent. Informally these are:

- An axiom insisting that you be willing to express preferences between simple dichotomized possible actions ($\{a, \text{not } a\}$);
- A *transitivity* axiom in which (for all actions a, a_1, a_2, a_3) $a \leq a$, and if $a_1 \leq a_2$ and $a_2 \leq a_3$ then $a_1 \leq a_3$; and
- An axiom based on the **sure-thing principle** (Savage, 1954): if, in two situations, no matter how the first comes out the corresponding outcome in the second is preferable, then you should prefer the second situation overall.

This puts $\leq$ on a sound footing for *qualitative* uncertainty assessment, but does not yet imply how to quantify—it's like being able to say that one thing weighs less than another but not to say by how much. To go further requires a fourth assumption, analogous to the existence of a set of *reference standards* (for example, an official kg weight, half-kg, and so on) and the ability to make arbitrarily precise comparisons with these standards:

- An axiom guaranteeing that for each outcome E there exists a **standard outcome** S (for instance, "idealized coin lands heads") such that $E \sim S$.

This framework implies the existence and uniqueness of a (personal) probability $P_{B:\text{you}}$ (abbreviated P), mapping from outcomes E to [0,1] and corresponding to the judgments in your definition of $\leq$, and a **utility function** U_{you} (abbreviated U; large values preferred, say), mapping from consequences c to the real line and quantifying your preferences.

This has all been rather abstract. Four concrete results arising from this framework may make its implications clearer:

- Bayes' original definition of personal probability is helpful in thinking about how to quantify uncertainty. Pretending that consequences are monetary (for instance, in US$), to say that $P_{B:\text{you}}(E) = p$ for some uncertain outcome E whose truth

value will be known in the future is to say that you are indifferent between (a) receiving $\$p \cdot m$ for sure (for some hypothetical amount of money $\$m$) and (b) betting with someone in such a way that you will get $\$m$ if E turns out to be true and \$0 if not.

- Any coherent set of probability judgments *must satisfy the standard axioms and theorems of a* **finitely additive** *probability measure*:
  - $0 \leq P(E) \leq 1$ and $P(E^c) = 1 - P(E)$;
  - $P(E_1 \text{ or } \ldots \text{ or } E_J) = \sum_{j \in J} P(E_j)$ for any finite collection $\{E_j,\, j \in J\}$ of disjoint outcomes;
  - $P(E \text{ and } F) = P(E) \cdot P(F)$ for any two independent outcomes (informally, E and F are *independent* if your uncertainty judgments involving one of them are unaffected by information about the other); and
  - *Conditional probability* has a natural definition in this setup, corresponding to the updating of your uncertainty about E in light of F, and with this definition $P(E|F) = \frac{P(E \text{ and } F)}{P(F)}$.

  Otherwise (de Finetti, 1937/1980) someone betting with you on the basis of your probability judgments can *make* **Dutch book** *against you*, which is to say this person can get you to agree to a series of bets that are guaranteed to lose you money. Thus coherent Bayesian probability obeys the same laws as with the classical and frequentist approaches.
- Nothing so far has said clearly what choice to make in a decision problem if you wish to avoid incoherence. If the outcomes were certain you would evidently choose the action that maximizes your utility function, but since they are not the best action must involve a weighing both of your probabilities for the uncertain outcomes and the utilities you place on their consequences. It is a direct implication of the framework here that the form this weighing should take is simple and clear:

  Maximization of expected utility (MEU): Given your utility and probability judgments, your decision-making is coherent iff for each action a_i, with associated uncertain outcomes $\{E_j, j \in J\}$ and consequences $\{c_j, j \in J\}$, you compute the *expected utility* $\text{EU}_i = \sum_{j \in J} U(c_j) P(E_j)$ and choose the action that maximizes $\{\text{EU}_i, i \in I\}$.

  This is the basis of rational choice theory in economics (e.g.,

von Neumann and Morgenstern, 1944). It has been shown (ref, 19xx) [if anybody can supply one of these references for me, I would be grateful] that in practice people sometimes act roughly like expected utility maximizers and sometimes they do not. Economists have a simple way out of this: utility is very hard to measure accurately, maybe there is nothing wrong with the theory, we just got their utility functions wrong. Or maybe the theory is incomplete: I recall an interesting talk given at Rand in 1989 by Howard Raiffa, one of the leaders of his generation in Bayesian decision theory, in which he was asked if he followed MEU in his own personal decision-making. He said, "Heck, no, the choices my wife and I were making [about which jobs to take, where to live, and so on] were far too important to leave to MEU." (!) He also said, though (and this accords with my own experiences), that he found laying out the *ingredients* of an MEU calculation—the possible actions, the values you would give to possible consequences, some rough idea of the relative likelihood of the uncertain outcomes—to be invaluable in making personal choices.

1.17 Computer scientists have terminology for an aspect of a computer program that some people regard as undesirable and others think is good: the former call it a *bug*, the latter a *feature*. For non-Bayesians having to specify a prior is a bug; for Bayesians it's a feature.

1.18 The idea of conjugacy is at its most general in the **exponential family** of parametric probability distributions:

> **Definition** (e.g., Bernardo and Smith, 1994): Given data $y = (y_1, \ldots, y_n)$ and a parameter vector $\theta = (\theta_1, \ldots, \theta_k)$, the sampling distribution $p(y|\theta)$ belongs to the *k-dimensional exponential family* if it can be expressed in the form
>
> $$p(y|\theta) = c\, f(y)\, g(\theta)\, \exp\left[\sum_{i=1}^{k} \phi_i(\theta)\, h_i(y)\right]. \qquad (1.47)$$
>
> In this case $\{\sum_{i=1}^n h_1(y_i), \ldots, \sum_{i=1}^n h_k(y_i)\}$ is the set of *sufficient* statistics for θ under $p(y|\theta)$.

As noted less formally in Section 1.6, $\{h_1, \ldots, h_k\}$ is sufficient

for θ under $p(y|\theta)$ if the likelihood $l(\theta|y)$ depends on y only through the values of $\{h_1, \ldots, h_k\}$.

I bring up the exponential family because, if the likelihood $l(\theta|y)$ is of the form (1.47), then in searching for a conjugate prior $p(\theta)$—that is, a prior of the same functional form as the likelihood—you can see directly what will work:

$$p(\theta) = c\, g(\theta)^{\tau_0} \exp\left[\sum_{i=1}^{k} \phi_i(\theta)\, \tau_i\right], \tag{1.48}$$

for some $\tau = (\tau_0, \ldots, \tau_k)$. With this choice the posterior for θ will be

$$p(\theta|y) = c\, g(\theta)^{1+\tau_0} \exp\left\{\sum_{i=1}^{k} \phi_i(\theta)\, [h_i(y) + \tau_i]\right\}, \tag{1.49}$$

which is indeed of the same form (in θ) as (1.48).

As a first example, with $S = \sum_{i=1}^{n} y_i$, the Bernoulli/binomial likelihood in (1.18) can be written

$$\begin{aligned} l(\theta|y) &= \theta^S (1-\theta)^{n-S} \\ &= (1-\theta)^n \left(\frac{\theta}{1-\theta}\right)^S \qquad (1.50) \\ &= (1-\theta)^n \exp\left[S \log\left(\frac{\theta}{1-\theta}\right)\right], \qquad (1.51) \end{aligned}$$

which shows (a) that this sampling distribution is a member of the exponential family with $k = 1$, $g(\theta) = (1-\theta)^n$, $\phi_1(\theta) = \log\left(\frac{\theta}{1-\theta}\right)$ (**NB** the basis of logistic regression), and $h_1(y_i) = y_i$, and (b) that $h_1(y) \equiv \sum_{i=1}^{n} h_1(y_i) = S$ is sufficient for θ. Then (1.48) says that the conjugate prior for the Bernoulli/binomial is

$$\begin{aligned} p(\theta) &= c(1-\theta)^{n\tau_0} \exp\left[\tau_1 \log\left(\frac{\theta}{1-\theta}\right)\right] \\ &= c\, \theta^{\alpha-1}(1-\theta)^{\beta-1} = \mathrm{Be}(\alpha, \beta) \qquad (1.52) \end{aligned}$$

for some α and β, as it should be.

For an example with $p > 1$, take $\theta = (\mu, \sigma^2)$ with the Gaussian likelihood:

$$
\begin{aligned}
l(\theta|y) &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2}(y_i-\mu)^2\right] \\
&= \sigma^{-n}(2\pi)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{n} y_i^2 \right.\right. \qquad (1.53) \\
&\qquad \left.\left. -2\mu\sum_{i=1}^{n} y_i + n\mu^2\right)\right].
\end{aligned}
$$

This is of the form (1.47) with $k = 2$, $c = (2\pi)^{-\frac{n}{2}}$, $f(y) = 1$, $g(\theta) = \sigma^{-n}\exp\left(-\frac{n\mu^2}{2\sigma^2}\right)$, $\phi_1(\theta) = -\frac{1}{2\sigma^2}$, $\phi_2(\theta) = \frac{\mu}{\sigma^2}$, $h_1(y_i) = y_i^2$, and $h_2(y_i) = y_i$, which shows that $[h_1(y) = \sum_{i=1}^{n} y_i^2, h_2(y) = \sum_{i=1}^{n} y_i]$ or equivalently $(\bar{y}, s^2)$ is sufficient for θ. Looking ahead a bit in the text, some very unpleasant algebra then demonstrates that (1.34) is conjugate for the Gaussian likelihood when both μ and σ^2 are unknown.

1.19 The Be(α, β) distribution converges to the Gaussian as $\alpha+\beta \to \infty$.

1.20 Named in honor of a letter sent by Oliver Cromwell to the elders of the Church of Scotland in 1xxx, at a moment in history when the Church leaders had already firmly made up their minds about [Dennis: please help me with details here, and a reference (Lindley, 19xx) where Cromwell's Rule is stated]; Cromwell wrote, "I beseech you, in the bowels of Christ, think it possible that you may be wrong."

1.21 The situation with a sharp null like H_0: $\mu = 405.25$ is less pleasant: for Bayesians to make sense of such a hypothesis, there must be a blob of probability exactly at 405.25 in the prior, making both the prior and posterior a funny mixture of discrete and continuous distributions. For example, somebody who bought into this framework might construct a prior by putting probability 0.4 precisely on $\mu = 405.25$ and spreading the other 0.6 out with a normal distribution scaled to integrate to 0.6. In practice your uncertainty about parameters like μ is typically considerably smoother than that, which would seem to call into question the whole enterprise of testing sharp nulls (but see Problem 1.5 for a counterexample). In general I find it better to pass right by the entire enterprise of hypothesis testing in fa-

vor of more informative posterior summaries such as (say) 90% central intervals.

1.22 It spoils a good story, but in fairness I have to report that on my 79th trip to the machine I got a free cup of "tea."

1.23 Carl Morris regards the two parts of the double expectation theorem as so important for applied statisticians that he refers to them as "Adam and Eve."

CHAPTER 2

Simulation-based computation

2.1 The need for Markov Chain Monte Carlo methods

Case study 2.1: *Estimation of species life-span from the stratigraphic fossil record.* One class of questions of particular interest to people like geologists and paleobotanists concerns the points in the remote past (a) when a given species first appeared on Earth and (b) when it became extinct. The standard way to estimate these points is to dig below the surface—for instance by taking a vertical, cylindrical core sample—and look for the first and last occurrences of the species in the fossil record, measured in (say) meters below ground level. By means of carbon dating and cross-referencing against "known" times of major past events, an approximate one-to-one correspondence can be established between distance below ground and the time scale of interest.

Table 2.1. *Observed locations, in meters below ground, of finds of 6 taxa of ammonites, from Macellari (1986) by way of Strauss and Sadler (1989).*

Name	n	Locations y_i
D. lambi (0)	14	484, 517, 533, 550, 690, 780, 850, 995, 1055, 1083, 1100, 1115, 1130, 1157
M. seymour. (1)	13	617, 634, 645, 667, 692, 707, 730, 748, 755, 772, 779, 793, 822
K. darwini (2)	13	608, 622, 650, 685, 693, 704, 725, 742, 757, 771, 780, 800, 820
G. gemmatus (3)	25	650, 700, 757, 785, 793, 800, 892, 911, 934, 961, 994, 1005, 1025, 1032, 1048, 1067, 1077, 1091, 1100, 1115, 1124, 1140, 1157, 1166, 1171
M. weddel. (4)	3	668, 700, 767
M. dens. α (5)	16	815, 900, 950, 967, 982, 1000, 1015, 1033, 1050, 1070, 1098, 1115, 1140, 1150, 1158, 1175

As an example of this sort of work, Tables 2.1 and 2.2 give data on the observed range of late Cretaceous ammonites—a kind of

mollusk that left behind flat, spiral fossil shells—from samples gathered by Macellari (1986) on Seymour Island in the Antarctic Peninsula (the data were digitized from Figure 1 in Strauss and Sadler, 1989[1]). Range information on 13 taxa of ammonite are given, and the data are in meters below the surface (I can't transform to time because Macellari and Strauss-Sadler don't say how to, but the late Cretaceous period ended about 70 million years ago). Most of the taxa have rather grand names—*Anagaudryceras seymouriense*, for instance—that are too big to fit into the tables except as abbreviations, and for "ease of subsequent reference" Strauss and Sadler number them, slightly curiously, from 0 to 12. Let's concentrate at first on one of the taxa, say *M. dens.* α, and denote the observed locations of fossil finds by $y_1, \ldots, y_n$. What sort of model would be appropriate for the y_i?

Table 2.2. *Observed locations, in meters below ground, of finds of 7 more taxa of ammonites (see Table 2.1).*

Name	n	Locations y_i
K. laurae (6)	8	900, 928, 950, 973, 992, 1008, 1024, 1160
A. seymour. (7)	10	908, 985, 1000, 1025, 1035, 1042, 1060, 1082, 1115, 1137
M. dens. γ (8)	11	935, 947, 1000, 1015, 1024, 1040, 1050, 1084, 1100, 1110, 1120
P. riccardi (9)	9	960, 990, 1000, 1016, 1033, 1050, 1074, 1115, 1132
M. dens. β (10)	8	967, 977, 990, 1000, 1030, 1048, 1066, 1080
P. lorvi (11)	3	988, 1032, 1115
P. ultimus (12)	4	1100, 1110, 1127, 1150

A model for species life-span data. The observations are conceptually continuous, so if my predictive uncertainty is exchangeable I must be in the realm of model (1.22): $F \sim p(F), (y_i|F) \overset{\text{IID}}{\sim} F$ for continuous F. Strauss (a statistician) and Sadler (a geologist) say that there is lots of evidence to support the assumption of randomness of fossil distribution, locally in space and time, for ammonites and many other species, and further they say this evidence indicates that a *Poisson process*[2] is a reasonable approximation. This makes the number n of finds of a given taxon at a particular location in a fixed interval in time follow a Poisson distribution, and it is a basic fact about the Poisson process (e.g., Ross, 1970) that conditional on n the finds are *uniformly* distributed through-

out the interval. So it would seem from the physical situation that, if this story holds, we don't have any model uncertainty about F: denoting the left and right endpoints of the true underlying range for the taxon in question by A and B, the model is evidently

$$\begin{aligned} (A, B) &\sim p(A, B) \\ (y_i | A, B) &\overset{\text{IID}}{\sim} U(A, B). \end{aligned} \tag{2.1}$$

Of course, this model needs to be checked before it is applied: for instance, under the Poisson process the gap lengths $l_i = (y_{i+1} - y_i)$ between successive finds should be IID exponential, and the l_k should exhibit no serial correlation. Strauss and Sadler, who examined these data, report little serial correlation and say that "gap lengths for ammonite data roughly follow an exponential distribution truncated below 8.5m, [which is] approximately the limit of resolution of Macellari's published [core sample] illustration."

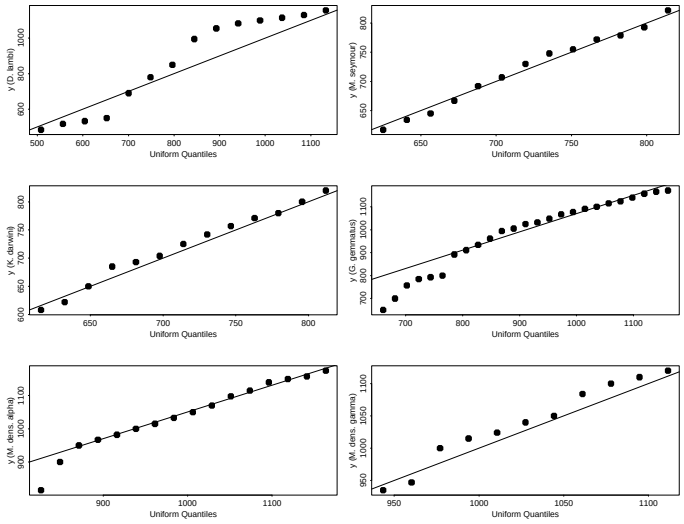

Figure 2.1. *Uniform qqplots for the ammonite data, showing the 6 taxa with the largest sample sizes.*

An even more direct way to check the distributional assumptions in model (2.1) is with uniform probability plots. Figure 2.1 presents such plots for the 6 taxa in Tables 2.1 and 2.2 with the largest sample sizes (it is hard to make much of a qqplot based on 3 or 4 points). Apart from a bit of wobble in the upper-left panel and a few left-tail outliers in several of the taxa (see Problem 2.1 for

a sensitivity analysis, exploring the effects of these points on the subsequent inferences), these plots are not desperately far from uniformity. I will go with model (2.1) in what follows, but I will differ a bit from Strauss and Sadler in that I am also interested in two other parameters: $\mu = \frac{B+A}{2}$, the center of the true range, and $\sigma = \frac{B-A}{2}$, a measure of the scale of this range. Reparameterized in this way the model is

$$\begin{aligned} (\mu, \sigma) &\sim p(\mu, \sigma) \\ (y_i|\mu, \sigma) &\overset{\text{IID}}{\sim} U(\mu - \sigma, \mu + \sigma). \end{aligned} \tag{2.2}$$

Computational strategies. Now it turns out that conjugate analysis of this model is possible if one of the two parameters is known but not if both are unknown (Problem 2.2). So, as with the NB10 t model mentioned in Section 1.8, conjugate analysis can only take us partway to the goal of {a fully Bayesian treatment of the broadest possible class of practically useful models}: a more flexible computing strategy is needed. In describing such a strategy I will digress for quite awhile, and then I'll return to the ammonite data in Section 2.5.

In this century people have known about the need for a better approach to computing for decades, of course, and (as I mentioned in Chapter 1) it was clear to Laplace way back in the 1770s that the integrals (1.30–1.32) in problems with multiple parameters could be immensely troublesome in general. With IID data $y = (y_1, \ldots, y_n|\theta)$ from a sampling distribution driven by a parameter $\theta = (\theta_1, \ldots, \theta_k)$ that is in most cases multivariate, three[3] main strategies, all of them with the goal of accurate approximate (rather than exact) results, have so far been developed to cope with this problem:

- **Asymptotic analysis** (e.g., Bernardo and Smith, 1994) relies on Central-Limit-Theorem-style results to approximate posterior distributions such as $p(\theta|y)$, $p(\theta_j|y)$, and $p(y_{n+1}|y)$ by suitable univariate and multivariate normal distributions[4]. With a few notable exceptions[5], for most Bayesians this was the leading supplement to conjugate analysis until the early 1980s.

  Asymptotic approximations work just about the way you might think they would, given what you know about the normal distribution: even with fairly small n this approach can produce reasonably accurate posterior summaries for parameters with

symmetric distributions on the whole real line, like μ in the Gaussian NB10 model (1.24). However, for parameters with skewed distributions and restricted ranges—such as θ in the Bernoulli/binomial model (1.11), which lives on (0,1), and σ^2 in (1.24), which lives on $(0, \infty)$—it is generally necessary to (a) **transform** the parameters to have support on all of $\mathbb{R}$ (for instance, by working with $\text{logit}(\theta) = \log\left(\frac{\theta}{1-\theta}\right)$ and $\log(\sigma^2)$, respectively), (b) do the normal approximation on the new scale, and (c) back-transform. Unfortunately, even after all of this fiddling about, in practice you may not get highly accurate results with small n.

- **Closed-form approximations** try to use higher-order asymptotic expansions than those on which standard asymptotic analysis are based to produce extremely accurate *small-sample asymptotics* (as some people put it). The leading example of this approach is the class of *Laplace approximations* (e.g., Tierney and Kadane. 1986), which I mentioned briefly in Chapter 1 and to which I will return in Chapter 8. Although invented by Laplace more than 200 years ago, the method seems to have languished with few practical applications until the 1980s. This approach works well with small k, particularly in conjunction with the kind of parameter transformation, or **reparameterization**, discussed above, but can lead to substantial numerical difficulties when k is large or the posterior distribution is multimodal.

- **Sampling-based approximations** try to take advantage of modern computing power, either (i) to approximate the integrals that arise in computing $p(\theta|y)$, $p(\theta_j|y)$, and $p(y_{n+1}|y)$ by Monte Carlo sampling experiments or (ii) to directly draw random samples from these distributions.

  - **Importance sampling** (e.g., Geweke, 1989) is an example of approach (i): for instance, if you want to calculate a posterior mean

$$E(\theta|y) = \int \theta \, p(\theta|y) \, d\theta \tag{2.3}$$

    and the integral in (2.3) is intractable, you could choose a density $g(\theta)$ that is everywhere positive, re-express (2.3) as

$$\int \theta \, p(\theta|y) \, d\theta \quad = \quad \int \left[\frac{\theta \, p(\theta|y)}{g(\theta)} \right] g(\theta) \, d\theta$$

$$= E_g\left[\frac{\theta\, p(\theta|y)}{g(\theta)}\right], \qquad (2.4)$$

take a large IID sample $(\theta_i, i = 1, \ldots, N)$ of points from $g(\theta)$, calculate $W_i = \frac{\theta_i\, p(\theta_i|y)}{g(\theta_i)}$, and approximate $E(\theta|y)$ by $\frac{1}{N}\sum_{i=1}^{N} W_i$. This method, which was extensively used in econometrics in the 1980s, requires considerable skill in the choice of the *importance sampling* density g, and (like many other approaches) runs into implementational and accuracy difficulties with large k.

- **Markov Chain Monte Carlo** (MCMC) methods (e.g., Gilks et al., 1996a) are the leading current example of approach (ii), and have been used extensively in statistics since the early 1990s with increasing success. Forerunners to this approach appeared in the statistics literature in the 1980s in the form of *data augmentation* (Tanner and Wong, 1987) and *sampling-importance-resampling* (Rubin, 1987), and MCMC methods were first widely popularized by Gelfand and Smith (1990), all of which makes it seem as though MCMC methods were not developed until the 1990s, but in fact the leading special cases of MCMC were introduced (a) in the 1950s (!) by physicists (Metropolis et al., 1953) whose work was unknown to the statistics community; (b) in the 1970s by a statistician (Hastings, 1970), whose efforts in generalizing Metropolis et al. went almost completely unnoticed; and (c) in the early 1980s by applied mathematicians (Geman and Geman, 1984) working in cognitive neuroscience, the generality of whose methods was not at first appreciated. (So much for cross-disciplinary collaboration and smooth historical sailing of important ideas.)

I am going to focus in this chapter—and in the rest of the book—on MCMC methods, because they appear to me (and to many others) to be the most promising general approach to Bayesian computation available at present. They can be highly computationally intensive (in other words, it can take minutes or even hours on your computer to get accurate answers), but I think it is fair to say that they have opened up the floodgates on applied Bayesian work since the early 1990s like no approach before them.

MCMC methods. The idea behind MCMC methods is simple and intuitive: I start out wanting to compute a probability den-

sity, for example $p(\theta|y)$, but then I notice after thinking about it for awhile that for many purposes *I would be just as happy to have a large random sample from $p(\theta|y)$ as to know its precise form*, because if I had the sample and it was big enough I could approximate its form, to a high degree of accuracy, with a histogram or kernel density estimator, and if I wanted to know its mean (say) I could approximate it by the sample mean. So the question becomes: can I figure out how to efficiently **simulate** a large number of random draws from $p(\theta|y)$?

This stage—the implementation of MCMC—is not so straightforward. In fact, it required a substantial bit of lateral thinking on the part of Metropolis et al. (the 1950s physicists). They said, in effect: suppose you could construct a **Markov chain**[6]—a stochastic process $\{\theta_t, t \geq 0\}$ of values unfolding in time t—with three properties:

- It should have the same **state space** (set of possible values) as θ;
- It should be easy to simulate from; and
- Its **equilibrium** (or **stationary**) **distribution**—the distribution from which samples from the chain eventually will be drawn, after it has been run for a long time—is $p(\theta|y)$.

If you could do this, you could run the Markov chain for a very large number of iterations, generating a huge sample $(\theta_t, t = 0, 1, \ldots)$ from the posterior, and then use *simple descriptive summaries* (means, SDs, correlations, histograms or density estimates) to extract any features of the posterior you want.

How to construct such a Markov chain—and make sure it does what you want it to—is the subject of the next several sections.

2.2 Hastings and Metropolis sampling

The most general MCMC method in wide use today is due to Hastings (1970), and I will look at it first (the methods due to Metropolis et al. and Geman-Geman are special cases, to be covered later). In effect Hastings said, OK, I am trying to build a Markov chain on θ starting at some **initial value** θ_0. Given that the chain has found its way to state θ_t at time t, all you need to know to characterize the chain (since it is Markov) is the probability distribution for where it will go at time $(t + 1)$. Hastings, following Metropolis

et al. but adding a new wrinkle (to be explained below), suggested the following way to generate θ_{t+1}:

- Choose something called a **proposal distribution (PD)** $f(\theta|\theta_t)$ for where to consider going next, given that you are at θ_t now, and sample a *candidate* point θ^* from this distribution.
- *Accept the move* to θ^* at time $(t+1)$ with probability

$$\alpha_H(\theta_t, \theta^*) = \min\left[1, \frac{p(\theta^*|y)\, f(\theta_t|\theta^*)}{p(\theta_t|y)\, f(\theta^*|\theta_t)}\right]; \tag{2.5}$$

  otherwise stay where you are. In other words, toss a Bernoulli coin with probability α_H of coming up heads—if you get heads, set $\theta_{t+1} = \theta^*$, otherwise set $\theta_{t+1} = \theta_t$.

He then proved the remarkable fact that with just about any PD f, the equilibrium distribution[7] for the Markov chain is $p(\theta|y)$, as desired.

Gilks et al. (1996b) note that the resulting algorithm is extremely easy to code:

Algorithm (Hastings, 1970, generalizing Metropolis et al., 1953). To construct a Markov chain whose equilibrium distribution is $p(\theta|y)$, choose a proposal distribution (PD) $f(\theta|\theta_t)$, define $\alpha_H(\theta_t, \theta^*)$ as in (2.5), and

```
Initialize θ_0 ; t ← 0
Repeat {
    Sample θ* ~ f(θ|θ_t)
    Sample u ~ U(0,1)
    If u ≤ α_H(θ_t, θ*) then θ_{t+1} ← θ*
        else θ_{t+1} ← θ_t
    t → (t+1)
}
```

(2.6)

Example: Gaussian with unknown σ^2 and known μ.

To see the Hastings algorithm in action, consider the simple model whose likelihood is specified by Gaussian draws with known mean but unknown variance, for instance applied to the NB10 data of Chapter 1 by pretending that we know μ:

$$\begin{aligned} \sigma^2 &\sim SI\text{-}\chi^2(\nu_p, \sigma_p^2) \\ (y_i|\sigma^2) &\overset{\text{IID}}{\sim} N(\mu, \sigma^2), \quad i = 1, \ldots, n. \end{aligned} \tag{2.7}$$

You can use the standard conjugate machinery to work out the right answer for $p(\sigma^2|y)$ in this model,

$$(\sigma^2|y) \sim SI\text{-}\chi^2\left(\nu_p + n, \frac{\nu_p\sigma_p^2 + ns_*^2}{\nu_p + n}\right), \tag{2.8}$$

where $s_*^2 = \frac{1}{n}\sum_{i=1}^n (y_i - \mu)^2$, so I'm not doing Hastings here because it's the only way to compute the posterior, but it's convenient to know the right answer so that I can compare the Hastings results with it. In this example $\theta = \sigma^2$, and I will take $\mu = 404.59$ (the NB10 sample mean) for illustration.

2.3 Practical implementation issues

If you look at the Hastings[8] algorithm (2.6) for awhile, you will see that three practical issues still need to be addressed in implementing it: picking the initial value θ_0; deciding how long to run the Markov chain and how to use the output to approximate $p(\theta|y)$; and choosing a PD $f(\theta|\theta_t)$. I will take a first crack at addressing all three of these issues in this section, in each case first by making some general comments and then coming back each time to the simple Gaussian model (2.7) above.

Choosing initial value(s) θ_0. The Hastings algorithm only mentions the need to choose a single initial value θ_0, and indeed in many problems one judicious choice of θ_0 is enough. If you are only going to pick one θ_0, the idea is to choose a value that is close to the center of the posterior distribution you are trying to simulate from—this will increase the chance that the Markov chain settles down into its equilibrium distribution quickly. A good θ_0 of this kind can come from anything simple that you may know about the posterior, for instance a decent frequentist estimate of θ like the **maximum likelihood estimate (MLE)**.

There is a potential danger in only choosing one θ_0, however. When the Markov chain is run it will wander around in θ–space over time t, and you would like to be sure that it has fully explored all regions of high posterior probability by the time you decide that the number of iterations you have looked at is enough. If the chain moves around freely, happily jumping all over the place, people say that it is **mixing well** (I will give some examples later in this section of poor mixing and good mixing). If (1) the posterior is multi-modal; (2) the particular MCMC implementation you are

currently using is mixing poorly; (3) you start the chain off near only one of the modes; and (4) you don't run it for very long, then you can see there is a real possibility you will never find the other mode(s).

There are two leading strategies for dealing with this problem: **simulated annealing** (SA; e.g., Geman and Geman, 1984) and **multiple highly dispersed initial values** (Gelman and Rubin, 1992). SA works by (a) thinking of mode-finding as like hill-climbing, with the hill(s) defined by $p(\theta|y)$, and (b) using a clever "non-greedy" strategy for iterative hill-climbing that sometimes is willing to go downhill to increase the chance of not getting stuck at a local maximum. The Gelman-Rubin plan is (a) to start up the chain at a number of wildly different θ_0 values and then (b) to see if it always converges to the same mode.

I like SA better than Gelman-Rubin because it turns out[9] that you can think of the Metropolis algorithm (see (2.21) below) as a special case of SA—so that you really only have to write one computer program to solve both the mode-finding and the posterior-sampling problems—but the Gelman-Rubin approach also has many fans (and I will give an example of it in Section x.x). Fortunately the problem of multiple modes generally only arises with HMs when you have used a highly informative prior that conflicts sharply with the likelihood, a situation you generally want to avoid in any case, so in what follows I will mostly deal with the question of initial values by choosing a single good θ_0.

Application to (2.7). The MLE for σ^2 in the simple Gaussian model (2.6) is s_*^2, the sample variance centered at the known $\mu = 404.59$, which in the NB10 data comes out 41.402, so that's what I'll use for σ_0^2 when I want to illustrate a good initial value below.

Choosing a convergence-monitoring strategy. This task in turn divides into two sub-tasks: how to decide when you've reached equilibrium, and how to monitor the output of the chain from that point onward to get posterior summaries of whatever accuracy you want.

- *Reaching equilibrium.* Think of the output of the chain as a time series indexed by the iteration counter t in (2.6). Equilibrium in this context is like **stationarity** of the time series, for which there are a variety of standard tests. I will cover this topic much more fully in Section 2.4 below; for now, pretend we have already solved the PD problem and consider Figure 2.2, which

was obtained using a particular PD I'll motivate below and an initial value that is far from the correct posterior mean in model (2.7) with the NB10 data.

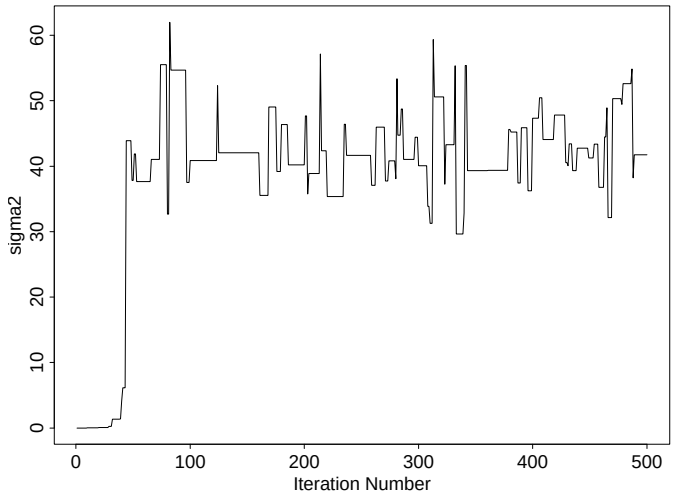

Figure 2.2. *Hastings output in model (2.7) with the NB10 data, using a PD of the form (2.15) and an initial value far from equilibrium.*

The output in this figure exhibits two undesirable features in MCMC sampling: it is not mixing very well—notice that there are fairly substantial periods (for instance, from about iterations 120 to 160) during which it does not move at all—and the beginning of the series was spent looking for equilibrium, which the sampler seems to have found at about iteration 50.

The first of these undesirable behaviors can be diagnosed by computing the **first-order autocorrelation**[10] (or **serial correlation**) of the series, which is about $\hat{\rho} \doteq +0.94$ in this case: a better-mixing chain would have a value of $\hat{\rho}$ much closer to 0. I will talk below about how to reduce serial correlation in Hastings samplers.

The simplest way to get around the second problem in Figure 2.2 is to *throw away the first 50 iterations* and then start monitoring the chain from that point on. People refer to an initial period which is discarded in this way as the **burn-in** period n_B for

the MCMC sampler. If the iterations are quick to compute and a good initial value is available, people often use a fairly standard value of n_B like 1,000 (or 5,000, just to be safe), and then increase n_B if the time series plot of the output shows that a larger burn-in period is needed.

- *Monitoring the chain to summarize the posterior accurately.* Suppose you are convinced that the sampler is in equilibrium after n_B burn-in iterations, which you have discarded. Then the basic idea for what to do next is (a) to run the chain for a further n_M monitoring iterations, creating what I will call the **MCMC data set**, and (b) to *approximate interesting features of the posterior distribution just by using simple descriptive summaries of this data set.*

Table 2.3. *Hypothetical MCMC data set in a model with parameter vector $\theta = (\gamma, \beta, \Delta)$, using n_B = 5,000 and n_M = 20,000. Here η is the derived quantity $\frac{\beta\Delta}{\gamma}$ and the y_t^* are draws from the predictive distribution for a new y.*

Iteration Number (t)	MCMC Phase	Sampled Values γ_t	β_t	Δ_t	$\eta_t = \frac{\beta_t \Delta_t}{\gamma_t}$	y_t^*
0	Initial Value	10.4	0.0762	−328	−2.40	54.2
1	Burn-in	11.7	0.0556	−359	−1.71	60.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n_B = 5000	Burn-in	9.26	0.0610	−274	−1.80	63.7
5001	Monitor	10.6	0.0804	−355	−2.69	49.9
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$n_B + n_M$ = 25000	Monitor	10.9	0.0793	−338	−2.45	58.1

Table 2.3 presents a hypothetical MCMC data set in a problem with data vector y and parameter vector $\theta = (\gamma, \beta, \Delta)$. (I will talk about the last two columns of this table in a few paragraphs.) Here let's suppose that 5,000 burn-in and 20,000 monitoring iterations are adequate to attain equilibrium and produce posterior summaries of sufficient accuracy for whatever you're doing. Then (a) if you'd like an estimate of the posterior mean of Δ, you can just calculate the sample mean of the

20,000 values $(-355, \ldots, -338)$ in rows 5,001 to 25,000 in the Δ column of the MCMC data set; (b) if you want a plot of the marginal posterior for γ, all you have to do is pass the 20,000 values $(10.6, \ldots, 10.9)$ in those same rows in the γ column of the MCMC data set to the histogram or kernel density trace function in S+ (or whatever your favorite data analysis package is); (c) if you'd like an estimate of the posterior correlation between β and Δ, you just calculate the sample correlation of the β and Δ columns in the monitoring part of Table 2.3; and so on. Estimates of (just about) anything you'd like to know about the posterior for θ—univariate, multivariate, whatever—are obtainable from the MCMC data set.

Notice that two of the difficult integration problems in Bayesian calculations with multivariate θ I mentioned in Chapter 1—*normalizing constants* (1.30) and *marginal posteriors* (1.31)—have disappeared with the MCMC approach: the normalizing constants cancel in the acceptance probabilities (look at the form of (2.5)), and sampling from the posterior instead of approximating the actual density makes marginalization trivial (if you want to know something that pertains only to β, say, then you simply ignore all the other columns in the MCMC data set).

It also turns out that the other two difficult integration problems from Chapter 1—computing *predictive distributions* (1.32) and *posteriors for functions of parameters* (1.33)—evaporate as well with MCMC. Concerning functions of parameters, you can convince yourself (Problem 2.3) that if $(\theta_t, t = n_B+1, \ldots, n_B+n_M)$ is a valid sample from the posterior for θ, then $[f(\theta_t), t = n_B + 1, \ldots, n_B + n_M]$ is a valid sample from the posterior for $f(\theta)$ for all reasonable f. This means that all you need to do is calculate $f(\theta)$ in each row of the MCMC data set and summarize as usual. The sixth column (counting from the left) of Table 2.3 illustrates this with the derived quantity $\eta = \frac{\beta\Delta}{\gamma}$ in the hypothetical model examined in that table.

As for prediction, recall from (1.32) that the predictive distribution for a new y—call it y^*, say—has the form

$$p(y^*|y) = \int p(y^*|\theta)\, p(\theta|y)\, d\theta. \tag{2.9}$$

The argument suggesting how to sample from this distribution with MCMC is in two parts.

(1) If you temporarily pretend the integral in (2.9) is a sum, I can probably convince you that summing the right-hand-side quantity $p(y^*|\theta)$ with index of summation θ and with respect to the density $p(\theta|y)$—in other words, computing $\sum_\theta p(y^*|\theta)\, p(\theta|y)$—is like taking a weighted average of the $p(y^*|\theta)$ values with weights (adding up to 1) given by $p(\theta|y)$, and if you let me pass from discrete to continuous θ in the right way and wave my hands a bit I can then probably convince you that in words *(2.9) says that the predictive distribution for y^* given y is a weighted average (or* **mixture***) of the sampling distributions $p(y^*|\theta)$ for y^* given θ, weighted by the posterior distribution $p(\theta|y)$ for θ given y.*

(2) I am getting considerably ahead of myself to bring it up here, since mixtures are the topic of Chapter 8, but it turns out (as we will see in that chapter) that *mixtures correspond directly to hierarchical models*. The HM suggested by the right-hand side of (2.9), in fact, has the simple form

$$\begin{aligned} \theta &\sim p(\theta|y) \\ (y^*|\theta) &\sim p(y^*|\theta). \end{aligned} \tag{2.10}$$

What (2.10) means in sampling terms is that

> To use MCMC to sample a y^* from $p(y^*|y)$, draw a θ from $p(\theta|y)$, say $\hat{\theta}$, and then sample y^* from $p(y^*|\hat{\theta})$.

Thus to draw from the predictive distribution of a new y in Table 2.3, for instance, you fill in each row from left to right, sampling $\hat{\gamma}_t$, $\hat{\beta}_t$, and $\hat{\Delta}_t$ (say), and then y_t^* is just a draw from the sampling distribution $p(y|\hat{\gamma}_t, \hat{\beta}_t, \hat{\Delta}_t)$ specified by the model featured in that table.

This all may sound too good to be true—all four major integration problems vanishing in one stroke—and in fact you may have developed a variety of something-for-nothing-style questions in reading the last few pages. The main question that occurs to me is

Q: Doesn't it say earlier this section that columns of the MCMC data set, when thought of as time series in t, often exhibit strong positive serial correlation, and doesn't that invalidate the MCMC data set as a summary of the posterior?

A: It's an interesting fact from time series (e.g., Anderson, 1971)

that if $(\theta_t, t = n_B+1, \ldots)$ is a stationary (correlated) process, then $\bar{\theta} = \frac{1}{n_M} \sum_{t=n_B+1}^{n_B+n_M} \theta_t$ is a consistent (as $n_M \to \infty$) estimate of $E(\theta)$, and similar results apply for SDs, correlations, and so on. So it's OK to simulate correlated draws from a distribution in summarizing features of that distribution, *as long as you get enough of them*, and that's where the amount of serial correlation comes in. You can show, for instance, that if θ_t is a (stationary) **autoregressive process**[11] of order 1 with mean μ, SD σ, and first-order serial correlation ρ (people abbreviate this $\theta_t \sim AR_1(\rho)$), then $\bar{\theta}$, as an estimate of μ, has standard error

$$SE(\bar{\theta}) = \frac{\sigma}{\sqrt{n_M}} \sqrt{\frac{1+\rho}{1-\rho}}. \tag{2.11}$$

If $\rho = 0$, which corresponds to an IID or **white noise** series, the SE has the usual $\frac{\sigma}{\sqrt{n_M}}$ form familiar to you from working with sample means of IID draws, but you can see that if ρ is close to $+1$ then the SE can become prohibitively large. For example, if $\rho = 0.9$, you would have to run the chain $\sqrt{\frac{1.9}{0.1}} \doteq 4.4$ times longer than if it had been white noise to get the same accuracy in estimating μ, and with $\rho = 0.995$ (which can happen) this multiplier is almost 20!

It turns out that, when considered as time series, MCMC samples for many quantities that you would want to monitor do behave a lot like AR_1 processes, so (2.11) is a useful formula in figuring out how long the chain should be run to achieve your accuracy goals. I will have more to say on this matter in Section 2.4.

Thinning the output. One more practical point: consider a situation in which you'd like your estimate $\bar{\theta}$ to have a high probability (95%, say) of being no more than (say) 0.1 from the correct posterior mean $\mu = E(\theta|y)$—in other words, you want

$$P(|\bar{\theta} - \mu| \leq d) = 1 - \epsilon \tag{2.12}$$

for $d = 0.1$ and $\epsilon = 0.05$. This is a sample size calculation, of the type that arises frequently when people design surveys, and the usual thing to do is to appeal to the Central Limit Theorem (CLT)—$\bar{\theta}$ is, after all, just a sample mean. Now it is another interesting fact from time series (e.g., An-

derson, 1971) that $AR_1(\rho)$ processes do obey the CLT (even though they are not IID unless $\rho = 0$), so—provided that your series looks like an AR_1 (I'll cover how to check this in Section 2.4)—the standard result from the sampling literature (e.g., Cochran, 1977), using the SE calculation in (2.11), yields the requirement that

$$n_M = \frac{\sigma^2 (1+\rho) \left[\Phi^{-1}(1-\frac{\epsilon}{2})\right]^2}{d^2 (1-\rho)}, \tag{2.13}$$

where σ is the SD of the θ_t and Φ is the usual standard normal CDF.

As is often the case with sample size calculations, the right side of (2.13) involves things you don't know, in this case ρ and σ. The natural thing to do here is to make a trial run of the sampler to estimate these quantities. Suppose you get $\hat{\rho} \doteq 0.89$ (in other words, your chain is not mixing very well) and $\hat{\sigma} \doteq 3.3$. Then (2.13) produces the rather sobering estimate $\hat{n}_M \doteq 79{,}500$, which we may as well round up to 80,000. Add a burn-in of (say) 5,000 iterations and we are up to 85K.

With the speed of today's machines (and the fact that next year's CPUs will probably be about twice as fast as today's), actually doing the 85,000 iterations may not be so bad, as far as clock time is concerned: if the trial run has shown that your computer can do about 50 iterations a second, for instance, 85K iterations works out to about 28 minutes, which might motivate a pleasant coffee break. But suppose θ has $k = 10$ components, and your worst-case $\hat{n}_M$ across all 10 parameters is 80,000. Then disk space starts to become an issue, as follows.

The MCMC data set will have 80,000 rows and $k + 1 = 11$ columns (including one for the iteration number). If you write it out to a character file for future data analysis, to obtain the posterior summaries of interest to you, in each row you'll need to allow 5 characters for the iteration counter and 7 characters for each parameter (given, say, 5 significant digits, a decimal point, and a space between each value). With $k = 10$ this comes out to $80{,}000 \cdot (5 + 7 \cdot 10) = 6$ megabytes of storage, and five or 10 runs like that can clog up your hard disk in no time.

So what most people would do in this situation is to make a long run of 80,000 but to only store every n_T-th row of the MCMC data set—this is called **thinning** the output of the chain. Here to hold the stored data set down to (say) 5,000 rows (which would only take up about 375K on disk), you would take $n_T = \frac{80000}{5000} = 16$.

In situations with extremely high serial correlation, I have sometimes needed to make monitoring runs of 1,000,000 or so iterations, storing every 200th, and I bet some readers of this book have made substantially longer runs than that, so thinning can be quite handy. It also acts to (greatly) reduce the serial correlation exhibited by the rows of the *stored* MCMC data set, although of course you still have to compute all n_M rows even if you store far less than n_M of them.

Choosing a PD $f(\theta|\theta_t)$. This is the hardest of the three practical tasks to pin down with any generality. Since the Hastings algorithm works for (just about) any PD, in fact there isn't just one Hastings solution to a given problem, there's an infinity of such solutions. The main goal in choosing $f(\theta|\theta_t)$ is getting a chain that mixes well, and nobody has (yet) come up with a sure-fire strategy for always succeeding at this task.

Having said that, here are two basic ideas that often tend to promote good mixing:

(1) Pick a PD that looks like a somewhat overdispersed version of the posterior you are trying to sample from (e.g., Tierney, 1994). Some work is naturally required to overcome the circularity inherent in this choice (if I knew $p(\theta|y)$, why would I be using this algorithm in the first place?).

(2) Set the PD up so that the expected value of where you are going to move to (θ^*), given that you accept a move away from where you are now (θ_t), is to stay[12] where you are now: $E_f(\theta^*|\theta_t) = \theta_t$. That way, when you do make a move, there will be an approximate left-right balance, so to speak, in the direction you move away from θ_t, which will encourage rapid exploration of the whole space.

The first chapter in Gilks et al. (1996a) has lots of good general ideas for choosing PDs. I will deal with this issue in the main body of this book principally by example, although in Section 2.6 and Appendix 2 Section 6 I will describe a fairly general strategy for

Metropolis sampling that employs one particular kind of generic PD.

Application to (2.7). Even if I didn't know the right answer (2.8) in this problem, a good place to begin in choosing the PD using idea (1) above—based on the form of the prior, and therefore the possible form of the posterior (given at least approximate conjugacy)—would be a scaled inverse χ^2 distribution: $f(\sigma^2|\sigma_t^2) = SI\text{-}\chi^2(\nu^*, \sigma_*^2)$ for some ν^* and σ_*^2. This distribution (Appendix 1) has density

$$p(\sigma^2|\nu^*, \sigma_*^2) = c\,(\sigma_*^2)^{\frac{\nu^*}{2}}(\sigma^2)^{-\left(\frac{\nu^*}{2}+1\right)} \exp\left(-\frac{\nu^*\sigma_*^2}{2\sigma^2}\right) \tag{2.14}$$

and mean $\frac{\nu^*}{\nu^*-2}\sigma_*^2$ for $\nu^* > 2$. To use idea (2) above, then, I can choose any ν^* greater than 2 that I want, and as long as I take $\sigma_*^2 = \frac{\nu^*-2}{\nu^*}\sigma_t^2$ I will have centered the PD at σ_t^2 as desired. So I will use

$$f(\sigma^2|\sigma_t^2) = SI\text{-}\chi^2\left(\nu^*, \frac{\nu^*-2}{\nu^*}\sigma_t^2\right). \tag{2.15}$$

This leaves ν^* as a kind of potential *tuning constant*—the hope is that I can vary ν^* to improve the mixing of the chain.

Section 1 of Appendix 2 contains some S+ functions to do Hastings sampling in this model with PD (2.15). Various details need filling in, as follows.

- *PD simulation.* S+ doesn't have a built-in function to sample from the scaled inverse χ^2 distribution, but it does generate random χ^2 draws nicely. As in Section 1.8, a bit of distributional manipulation bridges the gap: it turns out (Appendix 1) that

$$\sigma^2 \sim SI\text{-}\chi^2(\nu, s^2) \iff \frac{1}{\sigma^2} \sim \Gamma(\frac{\nu}{2}, \frac{\nu}{2}s^2). \tag{2.16}$$

  Now χ^2 distributions are just special gamma distributions—$\chi^2_\nu = \Gamma(\frac{\nu}{2}, \frac{1}{2})$—so if I could get the second parameter in (2.16) to be $\frac{1}{2}$ I'd be home. But the second parameter in gamma distributions is an *inverse scale* parameter, by which I mean that multiplying a $\Gamma(\alpha, \beta)$ random draw by c turns it into a $\Gamma(\alpha, \frac{\beta}{c})$ draw. So evidently

$$\frac{\nu s^2}{\sigma^2} \sim \Gamma(\frac{\nu}{2}, \frac{1}{2}) = \chi^2_\nu, \tag{2.17}$$

  meaning that to generate a random draw σ^2 from $SI\text{-}\chi^2(\nu, s^2)$

you just generate a draw d from χ^2_ν and compute $\sigma^2 = \frac{\nu s^2}{d}$. This explains the function `PD.sim`.

- *Log prior and log likelihood.* In the function `alpha` I compute the acceptance probability α_H in (2.5) by calculating $\exp(\log(\alpha_H))$, so I need to compute the log posterior and log PD densities at various points. The log posterior is in turn just the sum log(prior) + log(likelihood). The SI-χ^2 prior density was given in (2.14), except that here I am using ν_p and σ_p^2 in place of ν^* and σ_*^2; its logarithm is

$$\log\left[p(\sigma^2|\nu_p, \sigma_p^2)\right] = c - \left(\frac{\nu_p}{2} + 1\right)\log(\sigma^2) - \frac{\nu_p \sigma_p^2}{2\sigma^2}. \qquad (2.18)$$

  Note from the form of α_H that the constant c in (2.18) doesn't need to be computed—e^c cancels in the acceptance ratio—so I have used $c = 0$ in the function `log.prior`.

  The likelihood function in model (2.7) is a simple Gaussian:

$$\begin{aligned} l(\sigma^2|y) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(y_i - \mu)^2\right], \quad \text{so} \\ \log\left[l(\sigma^2|y)\right] &= c - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2, \quad (2.19) \end{aligned}$$

  and again for the same reason I have taken $c = 0$ in the function `log.lik`.

- *Log PD calculation.* I am using the PD (2.15), and in view of (2.14) the log of this density with $\sigma_*^2 = \frac{\nu^* - 2}{\nu^*}\sigma_t^2$ can, after a bit of simplification, be written

$$\begin{aligned} \log\left[p(\sigma^2|\sigma_t^2)\right] &= c + \frac{\nu^*}{2}\log(\sigma_t^2) - \left(\frac{\nu^*}{2} + 1\right)\log(\sigma^2) \\ &\quad - \frac{(\nu^* - 2)\sigma_t^2}{2\sigma^2}, \qquad (2.20) \end{aligned}$$

  and once again I used $c = 0$ in the function `log.PD`[13].

This may all seem like a lot of work, but in fact much of the process of creating a Hastings sampler is generic: for instance, the driver, acceptance probability, and log posterior functions require little change from problem to problem. You will probably find that once you have written one Hastings sampler from scratch, it doesn't take much effort to do another one. The same is true for the other

kinds of MCMC samplers as well (although there are various tricks to learn to get decent mixing in high dimensions).

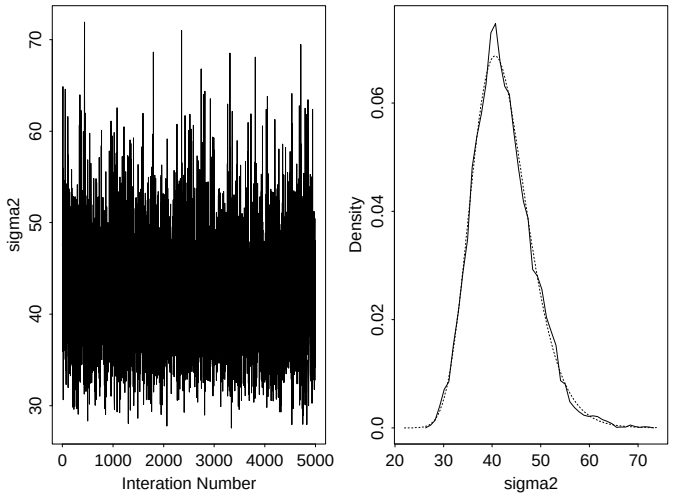

Figure 2.3. *Output of a Hastings sampler in the Gaussian model (2.7): Time series trace (left panel) and density trace (right panel) for* σ^2*. The solid curve on the right is based on a kernel density estimate from the 5,000 stored iterations; the dotted curve is the theoretical density.*

Hastings results for model (2.7). Figures 2.3 and 2.4 present results from applying the sampling strategy outlined above to the Gaussian model (2.7) with known mean and unknown variance, using the NB10 data for illustration. I chose $\nu^* = 20$ in specifying my PD (in Section 2.4 I'll justify this choice), and I used a burn-in of 1,000 starting from $\sigma_0^2 = s_*^2 = 41.402$, followed by a monitoring run of 40,000, storing every 8th iteration. For illustration, I set μ to $\bar{y} = 404.59$, and took $(\nu_p, \sigma_p^2) = (0.001, 41.402)$ (I will describe a more scientifically relevant prior in Section 2.6). This run took about 5.5 minutes using S+ on a 333Mhz machine[14], and yielded an acceptance rate of about 44%, which (as we will see in Section 2.4) leads to pretty good mixing (not far from best possible with a SI-χ^2 PD in this problem, in fact). Figure 2.3 plots the monitored iterations for σ^2 in two ways, and Figure 2.4 shows draws from the predictive distribution for a future y^*; in both cases the left panel is a time series trace of the 5,000 stored iterations, and the right

panel compares a kernel density trace based on the 5,000 draws with the theoretical density.

In both figures the time series traces look a lot more like white noise than Figure 2.2, which was produced by choosing $\nu^* = 5$ (leading to an acceptance rate of only about 20%); here the serial correlations for σ^2 and y^* were 0.03 and 0.00, respectively, compared with $\rho = 0.94$ back in Figure 2.2. And you can see that (apart from the vagaries of slightly undersmoothed kernel density traces) the MCMC distributions match their theoretical[15] counterparts well. The posterior means, SDs, and 95% central intervals for σ^2 and y^* from the MCMC output are $(42.2; 6.09; (32.0, 55.3))$ and $(404, 6.43, (392, 417))$, respectively, which pretty closely match their theoretical values $(42.2; 6.10; (32.0, 55.6))$ and $(405, 6.50, (392, 417))$.

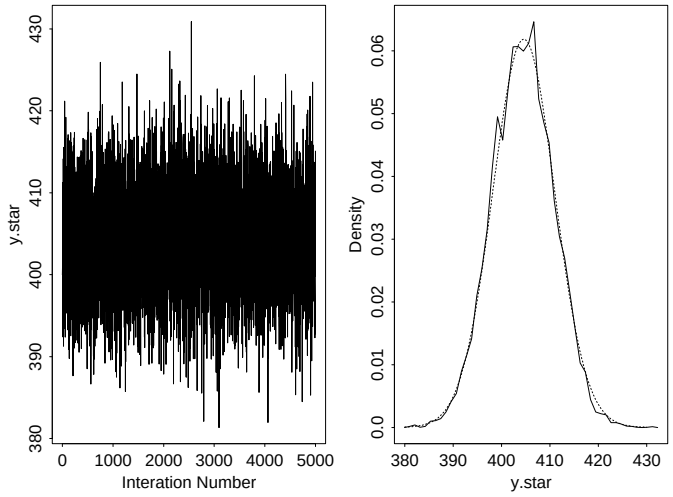

Figure 2.4. *Similar to Figure 2.3, but the quantity being monitored here is a future value y^* in model (2.7).*

S+ versus C. By virtue of its relatively friendly syntax, graphics capabilities, and interactive nature, S+ is an excellent environment in which to *develop* statistical software. Some of these features, however, act to hobble it sufficiently for MCMC-style calculations that it may not be the best environment in which to *run* such software. The reasons are as follows:

- The S+ people have implemented a design philosophy that in-

cludes a desire for their program to recover gracefully if interrupted in the middle of (essentially) any calculation. This is good in many ways but comes at a price: relatively poor *dynamic memory management*, particularly with explicit looping (of the kind that is unavoidable in MCMC: how can you avoid writing something like `for (i in 1:(n.burnin + n.monitor))`?). Naive versions of the `S+` programs in Appendix 2 routinely crash with the message `Unable to allocate dynamic memory`, even with only 2 or 3 parameters and (say) 20,000 monitoring iterations. In Appendix 2 I have implemented a trick I learned from Brian Ripley to overcome this problem; using this idea, the code can be run with far larger values of `n.monitor` without crashing.

- However, even with this memory-allocation trick `S+` MCMC code tends to run fairly slowly, because `S+` is an *interpreted*—rather than a *compiled*—language.

What is needed is to be able to throw a switch: to program in interpretive (interactive) mode while developing the code, and then switch over to compiled (more like old-fashioned batch) mode to get results. Some readers of this book are probably using other software environments, such as `GAUSS` and `MATLAB`, in which (I think; **manuscript readers**: is this correct?) such a switch may be thrown; I have stuck here with `S+`—which has no such switch—because it is the most widely used academic statistical software environment worldwide and because I'm familiar with it. But I evidently need a way to toggle between `S+` and a faster run-time environment.

One reasonably simple option is to convert your `S+` code into `C` once you have debugged it. Section 2 of Appendix 2, for instance, contains a `C` version of the `S+` Hastings sampler used above in model (2.7). When I began writing this book I was a complete `C` novice, but with the help of a few able graduate students it only took about an hour to translate the `S+` code for this example into a working `C` program (and experience—plus the generic nature of MCMC sampler coding—have cut this time substantially on subsequent problems. The `C` code takes a lot more lines, mainly because you have to supply your own random number generators, but these only need to be written once and stored in a library).

The point of spending this hour is run-time efficiency: the same code that took 5.5 minutes in `S+` on a 333Mhz machine only took *6.5 seconds* in `C`. I'm not claiming that the programs in Appendix 2

are best possible; I'm just noting that in this problem, a reasonably straightforward `C` implementation was **51** times faster than a reasonably straightforward `S+` implementation (and I have seen other MCMC applications where the advantage for `C` is more like 350 to 1). If I am only going to run my sampler a few times and I can get decent results with 40,000 iterations, then I'd rather wait 5 extra minutes for the `S+` code to finish than translate it into `C`, but what if (a) I want to write a simulation program in which random data sets are analyzed with MCMC, or (b) it takes more like 800,000 iterations to get accurate findings?

The MCMC moral seems to be: either find a statistical programming environment you're happy with—in which you can toggle back and forth between interpretive and compiled mode—or get somebody (and it may be turn out to be you) to convert your slow code into fast code[16].

Metropolis sampling. All of this section so far has been about implementing the Hastings (1970) sampler, which often requires a bit of ingenuity in the choice of proposal distribution (PD). There is, however, a simpler MCMC approach, due to Metropolis et al. (1953), as follows.

If you look at the form of the acceptance probability (2.5) in Hastings sampling, you will see that one particular assumption about the proposal distribution would make things easier: if the PD were *symmetric* in its two arguments, θ_t (which you will recall is where the chain is now) and θ^* (where the chain is thinking of going)—in other words, if $f(\theta^*|\theta_t) = f(\theta_t|\theta^*)$—then the ratio $\frac{f(\theta^*|\theta_t)}{f(\theta_t|\theta^*)}$ in (2.5) would cancel. This was the original idea Metropolis and his co-authors had almost 20 years before Hastings generalized it: Metropolis et al. suggested the use of symmetric PDs, and Hastings pointed out that PDs didn't have to be symmetric. Thus

Algorithm (Metropolis et al., 1953). Same as Hastings (2.5, 2.6), except that the proposal distribution (PD) $f(\theta|\theta_t)$ must be *symmetric*: in other words, it must satisfy $f(\theta^*|\theta_t) = f(\theta_t|\theta^*)$. In this case the acceptance probability simplifies to

$$\alpha_M(\theta_t, \theta^*) = \min\left[1, \frac{p(\theta^*|y)}{p(\theta_t|y)}\right]. \tag{2.21}$$

Notice that this automatically satisfies heuristic idea (2) in the

section earlier on choosing a good PD: symmetric proposals make unbiased moves (in a left-right sense along the number line).

Now why is (2.21) easier? Well, it often makes choosing the PD more straightforward: people just tend to implement Metropolis with their favorite symmetric distribution. One possibility, for instance, is to propose a $U(-c, c)$ move from where you are now (this was in fact what Metropolis et al. suggested); another possibility is to make a $N(0, \sigma_{PD}^2)$ move, where c and σ_{PD}^2 play the same *tuning constant* role that ν^* did in the Hastings example above. Notice that c and σ_{PD}^2 are both *scale factors* in their respective PDs: presumably you can tune them to get an acceptance rate that leads to good mixing.

If I propose $U(-c, c)$ moves from where I are now and you use $N(0, \sigma_{PD}^2)$ moves, our PDs would both have the feature that the probability of generating a move to θ^* from θ_t depends only on the *distance* $|\theta^* - \theta_t|$ between the target and current locations—in other words, in both cases there is a univariate density h such that $f(\theta^*|\theta_t) = h(|\theta^* - \theta_t|)$. People call an MCMC sampler based on such a proposal a **random-walk** Metropolis (or Hastings) algorithm (because the output of the sampler, examined only at the times when you actually do make a move, forms a *random walk* (e.g., Feller, 1968) in $\mathbb{R}^k$). These samplers are an important special case of the general Metropolis idea, since it turns out both that they are easy to program and they tend to have good MCMC convergence properties (e.g., Roberts 1996).

None of this sounds particularly applicable to the normal variance problem (2.7) I tackled above with Hastings, however: after all, it would look funny to propose a symmetric $U(-2, 2)$ (say) move from $\sigma_t^2 = 0.5$ (say), with a big chance of going negative, when everybody knows that σ^2 has to be positive. A moment's reflection indicates the way out of this problem, though: since $U(-c, c)$ and $N(0, \sigma_{PD}^2)$ moves would cause you to travel (in principle) all over the entire real line, the parameter you're sampling had better live on the whole line, too. That's easily enough accomplished with a parameter like σ^2: just work instead with $\lambda \equiv \log(\sigma^2)$, and monitor the function $\sigma^2 = e^\lambda$ in your MCMC draws.

Parameter transformation. On this line of reasoning I want to work with model (2.7) except re-expressed in terms of λ. If you look at the log likelihood (2.19) for this model, you will see that the only change needed there is to stick in λ every time you see $\log(\sigma^2)$, and e^λ everywhere σ^2 appears, so the new log likelihood

is

$$\log[l(\lambda|y)] = c - \frac{n}{2}\lambda - \frac{1}{2e^{\lambda}}\sum_{i=1}^{n}(y_i - \mu)^2. \tag{2.22}$$

The log prior, which will be based on (2.18), requires a bit more work, though: I have to put in the Jacobian for going from one parameterization to the other. With $\lambda = g(\sigma^2) = \log(\sigma^2)$, the usual result from probability, written on the log scale, is

$$\begin{aligned}\log\left[p(\lambda|\nu_p, \sigma_p^2)\right] &= \log\left\{p_{\sigma^2}\left[g^{-1}(\lambda)|\nu_p, \sigma_p^2\right]\right\} \\ &\quad + \log\left(\left|\frac{\partial g^{-1}(\lambda)}{\partial\lambda}\right|\right).\end{aligned} \tag{2.23}$$

Here $g^{-1}(\lambda) = e^{\lambda}$, $\left|\frac{\partial g^{-1}(\lambda)}{\partial\lambda}\right| = e^{\lambda}$, and from (2.18) $\log\left\{p_{\sigma^2}\left[g^{-1}(\lambda)|\right.\right.$ $\left.\left.\nu_p, \sigma_p^2\right]\right\} = c - \left(\frac{\nu_p}{2} + 1\right)\lambda - \frac{\nu_p\sigma_p^2}{2e^{\lambda}}$, so when all the dust settles

$$\log\left[p(\lambda|\nu_p, \sigma_p^2)\right] = c - \frac{\nu_p}{2}\lambda - \frac{\nu_p\sigma_p^2}{2e^{\lambda}}. \tag{2.24}$$

By way of proposal distribution I will use a Gaussian[17] centered at where I am now and with SD σ_{PD}, but what should this SD be? Heuristic idea (1) in the earlier section on choosing a PD suggested making the PD look like a somewhat overdispersed version of the posterior distribution, so maybe that would work here. You may recall from earlier study of maximum likelihood estimates (e.g., Lehmann, 1983) that when n is fairly large the MLE $\hat{\lambda}$ for λ should have approximate sampling distribution

$$\hat{\lambda} \dot{\sim} N(\lambda, \sigma_{\lambda}^2), \tag{2.25}$$

where $\sigma_{\lambda}^2 = \hat{I}_{\hat{\lambda}}^{-1}$ is the reciprocal of the observed Fisher information evaluated at the MLE,

$$\hat{I}_{\hat{\lambda}} = -\left(\frac{\partial^2}{\partial\lambda^2}\log[l(\lambda|y)]\right)_{\lambda=\hat{\lambda}}. \tag{2.26}$$

As long as the amount of prior information is small in relation to the data information, (2.25) implies that the posterior for λ will be approximately

$$(\lambda|y) \dot{\sim} N(\hat{\lambda}, \sigma_{\lambda}^2), \tag{2.27}$$

so heuristic idea (1) suggests in this case a PD of the form

$$f(\lambda|\lambda_t) = N(\lambda_t, \kappa\sigma_{\lambda}^2) \tag{2.28}$$

for some **scaling factor** $\kappa > 1$.

To implement this I still have to compute σ_λ^2. Differentiating the log likelihood (2.22) yields

$$-\frac{\partial^2}{\partial\lambda^2}\log\left[l(\lambda|y)\right] = \frac{\sum_{i=1}^n (y_i-\mu)^2}{2e^\lambda}, \tag{2.29}$$

which simplifies considerably when evaluated at the MLE: in this model $\hat{\sigma}^2_{\text{MLE}} = e^{\hat{\lambda}} = \frac{1}{n}\sum_{i=1}^n (y_i-\mu)^2$, so σ_λ^2 just reduces to $\frac{2}{n}$. Evidently I should use $\sqrt{\frac{2\kappa}{n}}$ as my proposal distribution SD, varying κ to get a decent acceptance rate.

I will talk more about how to choose the optimal κ in the next section; for now it is worth noting that intuitively there *ought* to be a best κ somewhere in the middle of its possible range, because

- If the proposal distribution SD is too big, when you do move you will make big moves, which is good, but you won't accept such a move very often (look at the form of the acceptance probability α_M in (2.21)), which is bad, and in the limit as the SD gets huge you will hardly ever move, leading to high autocorrelation and terrible mixing; and
- If the proposal distribution SD is too small, you'll accept the resulting moves frequently (which is good), but when you move you won't move very far (which is bad), and if you mentally let the SD go to 0 you'll see that again you have high autocorrelation and terrible mixing, because it will take the chain a very long time to flesh out the whole posterior.

Two qualitative conclusions emerge from this: proposal distribution SDs—and acceptance probabilities—somewhere in the middle of their possible ranges are best; and

> To increase the Metropolis or Hastings acceptance probability, you should *decrease* the proposal distribution SD.

This is why $\nu^* = 20$ worked better than $\nu^* = 5$ with the Hastings PD in the last section: the variance of a $SI\text{-}\chi^2(\nu^*, \sigma^2)$ distribution (Appendix 1) for $\nu^* > 4$ is $\frac{2\nu^*\sigma^4}{(\nu^*-2)^2(\nu^*-4)}$, which goes down as ν^* increases, so when $\nu^* = 5$ produced an acceptance rate that was too low (in other words, the proposal distribution SD was too big) the right thing to do was to increase ν^*.

Metropolis results for model (2.7). Section 3 of Appendix 2 contains a set of S+ functions to do Metropolis sampling in model (2.7) using the PD developed above. You can see how little needs to be changed from the Hastings code earlier in that appendix. To test

the code I used all of the same settings as with the Hastings results (among other things, this yielded a prior on λ that was equivalent to the previous highly diffuse prior on σ^2) and took $\kappa = 6$ (I'll explain this value in Section 2.4). The ensuing run took 8.8 minutes at 333Mhz (the extra time was almost entirely due to writing out three monitored quantities—λ, σ^2, and y^*—instead of two) and produced results for σ^2 and y^* that were identical to those from the Hastings approach, apart from MCMC sampling noise. Figure 2.5 is a plot of the results for λ, with the normal approximation (2.27) superimposed on top of the kernel density trace from the Metropolis output—you can see that $n = 100$ is sufficient to have reached asymptotic nirvana (to decent accuracy, at least) with this data set.

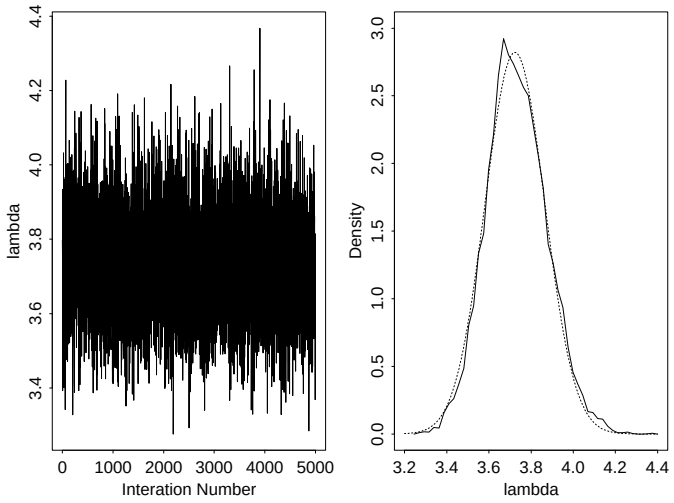

Figure 2.5. *Output of a Metropolis sampler in the Gaussian model (2.7): Time series trace (left panel) and density trace (right panel) for* $\lambda = \log(\sigma^2)$. *The solid curve on the right is based on a kernel density estimate from the 5,000 iterations; the dotted curve is the normal approximation (2.27), based on the MLE, to the posterior density.*

2.4 MCMC monitoring and convergence diagnostics

I have been promising for some time now to discuss methods for figuring out whether the chain is in equilibrium yet, and how long to run it after it has reached equilibrium. This has been an active

research area in the last 10 years (e.g., Brooks and Roberts, 1995; Cowles and Carlin, 1996) and will certainly continue to develop, but a number of useful methods have already been documented, as follows.

The first thing I often do is make a graph like Figure 2.6, which for want of a better name I will call an *MCMC 4-plot*. To create this picture, I reran my Hastings sampler from Section 2.3 on the parameter σ^2 with a far-from-optimal value of ν^*—2.5—and used a burn-in of 1,000 and a monitoring run of 5,000. This took about 1 second at 333Mhz and produced (on purpose) an abysmally low acceptance rate of only about 7%.

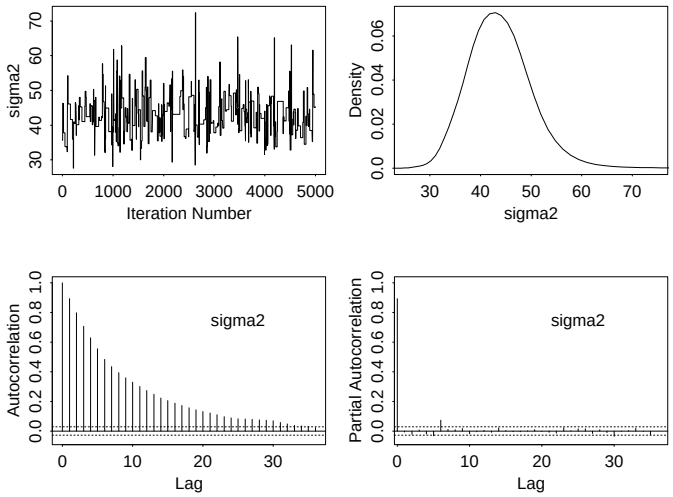

Figure 2.6. *MCMC 4-plot of results from the Hastings sampler in Section 2.3 on* σ^2*, with* $n_B = 1000, n_M = 5000$*, and* $\nu^* = 2.5$.

The upper left panel in Figure 2.6 is a time series trace of the 5,000 monitoring iterations. Here it shows pretty bad mixing: note that (a) the chain did not move at all for significant periods, and (b) if you ran a kind of mental "local smoother" through the plot, trying to estimate the mean of the time series near any given point, it would show a lot of wavy behavior, whereas white noise would just look like random fluctuations around a horizontal line. The upper right panel is a (considerably smoothed) kernel density trace of the monitored iterations[18], and (in spite of the poor mixing) already looks a lot like the correct answer (compare with the right panel in Figure 2.3).

The lower left and right panels in Figure 2.6 are plots of the autocorrelation and **partial autocorrelation**[19] functions (ACF and PACF; e.g., Box and Jenkins, 1976) for the 5,000 correlated draws from the posterior for σ^2. If you have studied the time-domain approach to time series, you will recognize that these plots are exhibiting the textbook behavior of an AR_1 series with a first-order autocorrelation of about 0.9: the PACF has one big spike at lag 1 of size 0.9, and the rest of the spikes are negligible (the dotted lines are two standard error traces around 0 for judging which spikes are worth taking notice of), and the ACF shows a slow geometric-style decay from an autocorrelation of 0.9 at lag 1 to values near 0 out around lag 35 (or even later).

Taken together the panels of the 4-plot show a chain that could well be in equilibrium (I don't see any vertical drift in the time series trace) but that likely needs to be run for considerably longer than 5,000 iterations to get accurate posterior summaries (because of the high serial correlation). With S+ handy, the easiest way to figure out how much longer is to invoke the MCMC diagnostic routines in a package called CODA.

The CODA diagnostics. CODA (Best et al., 1995) is a set of S+ functions available free on the web or by anonymous ftp from the Medical Research Council Biostatistics Unit in Cambridge, UK (see Appendix 2 for details on how to get the code). These functions offer six different kinds of MCMC convergence diagnostics, some of which I will now describe.

The simplest things you can get out of CODA are numerical estimates of the autocorrelation functions for each monitored quantity and the degree of cross-correlation exhibited by all the different time series you have generated, taken pairwise. I ran CODA on the Hastings output illustrated in Figure 2.6, obtaining the results in Tables 2.3–2.6 (I also monitored the predictive distribution for a future observation y^*). Section 1 of Appendix 2 gives an S+ function called preCODA to prepare the MCMC data set for reading by CODA.

Autocorrelations. Table 2.4, for example, gives the autocorrelations for σ^2 and y^* and the degree of cross-correlation between them. The ACF for σ^2 is a numerical match to the upper left panel in Figure 2.6, and shows the slow decline in the autocorrelations (you have to go all the way out to nearly lag 50 with this choice of ν^* to get close to IID sampling). You can also see that there

is little correlation between σ^2 and y^*, and that—even with this ν^*—y^* looks like white noise.

Table 2.4. *Autocorrelations and cross-correlations for the Hastings output illustrated in Figure 2.6.*

```
LAGS AND AUTOCORRELATIONS WITHIN EACH CHAIN:
============================================

Iterations used = 1:5000
Thinning interval = 1
Sample size per chain = 5000

-+--------+------------+----------------------------------------+-
 | Chain  |  Variable  |  Lag 1     Lag 5     Lag 10    Lag 50   |
 | =====  |  ========  |  =====     =====     ======    ======   |
 |        |            |                                        |
-+--------+------------+----------------------------------------+-
 | h1     |  sigma2    |  0.89400   0.55500   0.33000   0.02100  |
 |        |  y.star    |  0.00429   0.00159   0.01030   0.00608  |
 |        |            |                                        |
-+--------+------------+----------------------------------------+-

            CROSS-CORRELATION MATRIX:
            =========================

            Chain: hastings1
            ================

            -+----------+------------------------+-
             | VARIABLE |   sigma2    y.star     |
             | ======== |                        |
             |          |                        |
             | sigma2   |    1.00000             |
             | y.star   |   -0.00716   1.00000   |
             |          |                        |
            -+----------+------------------------+-
```

Geweke and Heidelberger-Welch. Two other useful MCMC diagnostics produced by CODA are due to Geweke (1992) and Heidelberger and Welch (1983). Geweke proposed a simple method based on time series ideas. He reasoned that, if the chain were in equilibrium, the means of the first (say) 10% and the last (say) 50% of the iterates should be nearly equal. So to calculate his diagnostic he just does a Z-test of the hypothesis of equality of these two means, and reports the resulting Z scores (on the usual standard normal scale), one for each monitored quantity. Thus Geweke Z-scores a lot bigger than (say) 2 in absolute value indicate that the mean level of the time series is still drifting, even after whatever burn-in you have already done, and you should rerun your chain with a longer burn-in before starting your monitoring. Here (Ta-

ble 2.5) there is perhaps a hint that a longer burn-in would have been useful for σ^2.

Table 2.5. *Geweke diagnostics for the Hastings output illustrated in Figure 2.6.*

```
GEWEKE CONVERGENCE DIAGNOSTIC (Z-score):
========================================

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

-+----------+-----------------+-
 | VARIABLE |   hastings1     |
 | ======== |   =========     |
 |          |                 |
 | sigma2   |   -1.760        |
 | y.star   |    0.626        |
 |          |                 |
-+----------+-----------------+-
```

Heidelberger and Welch (1983) proposed a diagnostic approach that uses the Cramer-von Mises statistic[20] to test for stationarity. CODA's implementation of the Heidelberger-Welch approach goes like this:

- If overall stationarity fails for a given quantity being monitored, CODA discards the first 10% of the series for that quantity and recomputes the Cramer-von Mises statistic statistic, continuing in this manner until only the final 50% of the data remain.
- If stationarity still fails with the last half of the data, then CODA reports overall failure of the stationarity test.
- CODA also computes a *half-width* test, which tries to judge whether the portion of the series that passed the stationarity test is sufficient to estimate the posterior mean with a particular default accuracy. The idea is to use time-series methods to estimate the standard error of the mean of the MCMC draws and then compute half of the width of the resulting frequentist 95% interval estimate for this mean (namely, 1.96 times the standard error). If this is less than the default tolerance (in CODA, set to ϵ times the sample mean, for $\epsilon = 0.1$), the retained portion of the chain passes the half-width test. **NB** (1) This is not very stringent—if you use this test in CODA, you may well wish to make ϵ smaller. (2) The half-width test is directly related to equations (2.11–2.13).

Here, as Table 2.6 indicates, even this rather poorly-mixing chain

gets over the Heidelberger-Welch hurdles with no problem, reinforcing the visual impression of no troubles with stationarity in Figure 2.6.

Table 2.6. *Heidelberger-Welch diagnostics for the Hastings output illustrated in Figure 2.6.*

```
HEIDELBERGER AND WELCH STATIONARITY AND INTERVAL HALFWIDTH TESTS:
================================================================

Precision of halfwidth test = 0.1

Chain: hastings1
================
```

VARIABLE	Stationarity test	# of iters. to keep	# of iters. to discard	C-vonM stat.
sigma2	passed	5000	0	0.321
y.star	passed	5000	0	0.105

VARIABLE	Halfwidth test	Mean	Halfwidth
sigma2	passed	42.9	0.499
y.star	passed	404.0	0.167

Raftery-Lewis. The CODA diagnostic I find the most useful is due to Raftery and Lewis (1992). Given the output of an MCMC sampler, Raftery and Lewis address the question of how long to monitor the chain, and in doing so they recognize that this in turn should be based on the answer to another question: how accurate do you want the posterior summaries to be? So they ask you, the user, to specify three things:

- Which quantiles of the marginal posteriors are you most interested in? Usually the answer is the 2.5% and 97.5% points, since they are the basis of a 95% interval estimate.
- With what minimum probability do you want to achieve your accuracy goals? The default is 95%.
- How accurate would you like the estimated quantiles of interest to be? This, in turn, can be measured in two different ways: taking the 0.025 percentile as an example, you could either specify

that the quantile q corresponding to the 0.025 point in the CDF be accurate to a given tolerance, or that the area to the left of the reported quantile be within a given margin of 0.025. Raftery and Lewis have opted for the latter (which does not seem to me to be the more natural choice), and the CODA default on this scale is 0.005—in other words, the default tries to set it up so that if you report a nominal 95% interval by quoting the 0.025 and 0.975 points in the MCMC output, the actual posterior probability of your interval will be between 0.94 and 0.96.

Here is how their methods works. Given a particular quantity θ that you have monitored and a particular quantile q of interest in θ's distribution, Raftery and Lewis dichotomize the output of the chain, replacing that output by a binary time series that is 1 if $\theta_t \leq q$ and 0 otherwise. They then assert that this binary chain should be approximately Markovian, and use standard results for two-state Markov chains to estimate how long the chain should be run to achieve the desired accuracy for the chosen quantile.

Table 2.7. *Raftery-Lewis diagnostics for the Hastings output illustrated in Figure 2.6.*

```
RAFTERY AND LEWIS CONVERGENCE DIAGNOSTIC:
=========================================

Quantile = 0.025
Accuracy = +/- 0.005
Probability = 0.95

Chain: hastings1
================

-+----------+-----------------------------------------------------+-
 |          |  Thin    Burn-in  Total   Lower bound  Dependence   |
 | VARIABLE |  (k)       (M)    (N)        (Nmin)    factor (I)   |
 | ======== |  ====    =======  =====   ===========  ==========   |
 |          |                                                     |
 | sigma2   |  1       58       68727   3746         18.3         |
 | y.star   |  1       2        3866    3746         1.03         |
 |          |                                                     |
-+----------+-----------------------------------------------------+-
```

As you can see from Table 2.7, Raftery and Lewis actually provide three kinds of estimates, in columns 2–4: what thinning ratio to use, how much additional burn-in would be useful the next time you run the chain, and the required length of (burn-in + monitoring) period—let's call it $\hat{n}_{RL}$—to achieve your accuracy goals. Column 5 shows the length of the chain required to meet those

goals if it had been white noise, and column 6 reports the ratio of columns 4 and 5, which Raftery and Lewis call the *dependence factor* $\hat{I}$.

I don't usually find the Raftery-Lewis thinning-ratio and extra-burn-in recommendations very useful (I tend to decide on thinning requirements based on storage considerations, and the recommended extra burn-in is usually trivially small). The column called `Total (N)`—$\hat{n}_{RL}$—is interesting, though: its punchline in this case, having used a proposal distribution with a deeply suboptimal value of ν^*, is that I need to rerun the chain for almost 70K iterations, to achieve the Raftery-Lewis default accuracy goals for the endpoints of my 95% interval for σ^2.

Of course, with the `C` program in Appendix 2 this is not hard: increasing n_B to 5,000 and n_M to 70,000 and storing every $n_T = 14$th iterate only takes about 11 seconds at 333Mhz. This passes all tests and reduces the first-order serial correlation of the *stored* iterates to 0.288, yielding a new $\hat{n}_{RL}$ of 6,756, which is more than the effective sample size of the new run (5,000). This sort of thing often happens—the first estimate of $\hat{n}_{RL}$ is a bit conservative because it isn't based on enough data yet. Now $6756 \cdot 14 \doteq 95$K, so I reran the chain for 100K iterates, storing every 20th, producing a new $\hat{n}_{RL}$ of 5391 (!), at which point I decided that the resulting answers would be close enough for government work (no, actually I kept on, out of curiosity, and I had to go all the way out to 140K, storing every 28th, before I had 5,000 σ^2 draws that passed the default accuracy goals. The moral seems to be that $\hat{n}_{RL}$ may well be biased on the low side when based on a modest number of draws).

Optimizing the proposal distribution. Now that $\nu^* = 2.5$ has proven itself to be a rotten tuning constant for the Hastings proposal distribution, the availability of `CODA`'s $\hat{n}_{RL}$ facility makes me wonder what the optimal ν^* is[21]. Table 2.8, which is based on a series of runs with $n_B = 5{,}000$, $n_M = 400{,}000$, and $n_T = 1$, investigates this question. You can see that as ν^* increases from its smallest value in the table, so does the (estimated) acceptance probability $\hat{\alpha}$, but both the autocorrelation $\hat{\rho}$ and the default $\hat{n}_{RL}$ reach a minimum in the middle, around $\nu^* = 20$–30. Thus, as we saw by reasoning qualitatively in Section 2.3, the best acceptance probability will be somewhere in the middle (in this case around 0.44–0.51).

As an alternative to Raftery-Lewis, a different but related way

to figure out what n_M should be is based on equation (2.13), which addresses an accuracy goal for the posterior mean rather than for percentiles: how many draws should you take so that the estimate of the posterior mean of θ you quote is correct to within a tolerance d with probability $(1-\epsilon)$? For instance, the posterior mean of σ^2 in model (2.7) with the NB10 data and a diffuse prior, based on the 5,000 draws shown in Figure 2.3, is 42.2, with a posterior SD of 6.1. How much longer than 5,000 should I have run the chain to confidently quote all three of the significant figures in the estimate 42.2?

Table 2.8. *Optimal choice of ν^* in the Hastings proposal distribution for the normal variance model (2.7). SSIF = sample size inflation factor (see text).*

ν^*	$\hat{\rho}$	$\hat{\alpha}$	$\hat{n}_{RL}$ (thousands)	SSIF $\left(\frac{1+\hat{\rho}}{1-\hat{\rho}}\right)$
2.5	0.903	0.068	94.6	19.6
5.0	0.743	0.202	31.9	6.78
10.0	0.652	0.320	20.7	4.75
20.0	0.625	0.443	18.1	4.33
25.0	0.632	0.482	17.8	4.43
30.0	0.643	0.513	17.0	4.60
40.0	0.667	0.563	19.0	5.01
50.0	0.688	0.598	19.5	5.41
100.0	0.779	0.698	23.8	8.05
500.0	0.928	0.858	50.3	26.8

The answer to this question in turn depends on ν^*. The part of equation (2.13) that is sensitive to the proposal distribution is the ratio $\left(\frac{1+\hat{\rho}}{1-\hat{\rho}}\right)$, which I have termed the *sample size inflation factor (SSIF)* and listed in the last column of Table 2.8. This is the amount that n_M needs to be multiplied by to satisfy accuracy goal (2.12), compared with its required value under IID sampling. With the best ν^*—the value that minimizes $\hat{\rho}$, namely $\nu^* \doteq 20$—the SSIF for Hastings sampling in this problem with a proposal distribution of the form (2.15) is 4.33. Now for the final 2 in 42.2 to be right, I need $d = 0.05$, and if I pick 95% as the desired level of confidence, equation (2.13) says that n_M would have to be $\frac{6.1^2 \ 1.96^2}{0.05^2} \cdot 4.33 \doteq 248$K! When you contrast this with what people often do (burn-ins of 1–5K followed by monitoring runs of 5–10K are common), it would seem that most of us (myself included,

until I made this calculation) do not run our samplers for as long as perhaps we should[22].

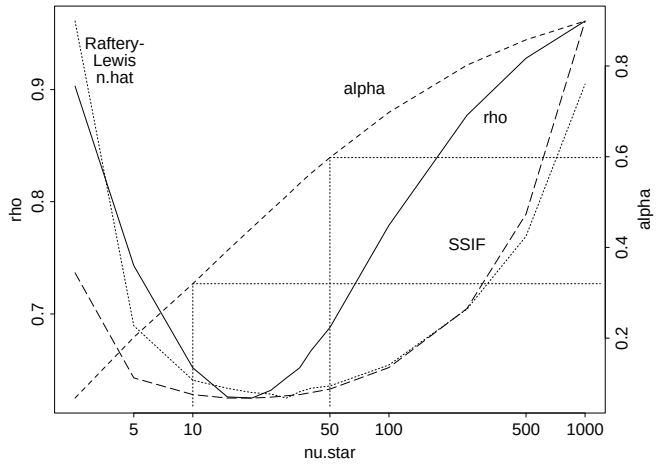

Figure 2.7. *A graphical version of Table 2.8: $\hat{\rho}$ (solid line), $\hat{\alpha}$ (small-dashed line), $\hat{n}_{RL}$ (dotted line), and the SSIF (large-dashed line) are plotted against ν^*.*

Figure 2.7 endeavors to wrap all of this up in one plot. The horizontal scale, expressed logarithmically, is ν^*; the left-hand vertical scale is ρ; the right-hand vertical scale is α; and columns 2–5 of Table 2.8 are plotted against column 1 in that table. You can see that, while the $\hat{n}_{RL}$ and SSIF criteria do not quite agree on the optimal ν^*, the region of near-optimality is broad, extending from about 10 to about 50, and this corresponds in turn to a broad range of target values for the acceptance probability, in this case from about 0.3 to about 0.6. Gelman et al. (1996) report something similar in the case of a Gaussian model with unknown mean rather than variance: with a Gaussian PD and a criterion that is different yet again from Raftery-Lewis and SSIF, they find the best α to be about 0.4, with values from 0.3 to 0.6 not far from optimal.

Optimal Hastings versus optimal Metropolis. The calculations in the previous subsection can be repeated with the Metropolis sampler on $\log(\sigma^2)$ introduced earlier. Table 2.9, which summarizes the results, demonstrates behavior similar to that seen for Hastings, with the following exceptions:

- As noted above, κ is a scale factor, whereas ν^* is inversely related to scale, so as κ increases the acceptance probability goes down.
- Metropolis appears a bit better-behaved than Hastings in this example, at least as far as $\hat{n}_{RL}$ is concerned (in contrast, they are about equally good—or bad—when measured by serial correlation and the SSIF).

The optimal κ is around 6, corresponding to an acceptance rate of about 0.44, although the region of near-optimality is again quite flat. This also agrees pretty well with what Gelman et al. (1996) found—in their problem the optimal multiplier, on the variance scale, was about 5.8.

Table 2.9. *Optimal choice of κ in the Metropolis proposal distribution for the normal variance model (2.7).*

κ	$\hat{\rho}$	$\hat{\alpha}$	$\hat{n}_M$ (thousands)	SSIF $\left(\frac{1+\hat{\rho}}{1-\hat{\rho}}\right)$
0.3	0.902	0.830	45.1	19.4
1.0	0.781	0.705	24.4	8.13
2.0	0.700	0.609	17.7	5.67
3.0	0.660	0.547	15.1	4.88
6.0	0.630	0.438	13.2	4.41
8.0	0.634	0.393	13.8	4.46
10.0	0.644	0.360	14.7	4.62
30.0	0.732	0.224	20.4	6.46
100.0	0.837	0.126	36.4	11.3

2.5 Gibbs sampling

Case study 2.1 (continued). I am finally ready to return to the geological example with which the chapter began. Recall that the statistical setup was

$$\begin{aligned} (A, B) &\sim p(A, B) \\ (y_i | A, B) &\overset{\text{IID}}{\sim} U(A, B), \quad i = 1, \ldots, n. \end{aligned} \tag{2.30}$$

for observed locations y_i, in meters below ground, of finds of one taxon of ammonites—a particular kind of fossil mollusk (Tables 2.1 and 2.2). Here A and B are the true lifespan limits of this taxon; let's pick *M. dens.* α as an example.

Bayesian inference in this problem involves the usual two steps,

beyond figuring out the likelihood: I need to specify a scientifically-reasonable prior, and I need to figure out how to compute the marginal posteriors, for instance $p(A|y)$. Actually, there is another task, as well: I can consider reparameterizing, for instance (as I did earlier) by defining $\mu = \frac{B+A}{2}$, the center of the true range, and $\sigma = \frac{B-A}{2}$, a measure of the scale of this range, and re-expressing the model as

$$\begin{array}{rcl} (\mu, \sigma) & \sim & p(\mu, \sigma) \\ (y_i|\mu, \sigma) & \overset{\text{IID}}{\sim} & U(\mu - \sigma, \mu + \sigma). \end{array} \tag{2.31}$$

I find it easier to think about things in this location-scale parameterization, so that's how I'll approach the problem here, regarding $A = \mu - \sigma$ and $B = \mu + \sigma$ as derived quantities to be monitored rather than elicited.

As far as the prior goes, the main thing that was known about these ammonites prior to Macellari (1986), from which the data in Tables 2.1 and 2.2 were taken, is that they were from the late Cretaceous period. On the meters-below-ground scale on Seymour Island in the Antarctic Peninsula, the source of the data, this period corresponded roughly to the range from $L = 400$m to $H = 1700$m. For any given taxon, this implies a prior in which μ can be pretty much anywhere between L and H, with no particular values favored, and σ is quite free too, subject to the restriction that $L < \mu - \sigma$ and $\mu + \sigma < H$. Rearranging these two inequalities and insisting that $\sigma > 0$, in keeping with a scale parameter, gives the prior

$$\begin{array}{rcl} \mu & \sim & U(L, H) \\ (\sigma|\mu) & \sim & U[0, \min(\mu - L, H - \mu)]. \end{array} \tag{2.32}$$

Here

$$\min(\mu - L, H - \mu) = \left\{ \begin{array}{cccc} \mu - L & \text{for} & L < \mu < \mu_* \\ H - \mu & \text{for} & \mu_* < \mu < H \end{array} \right\}, \tag{2.33}$$

where μ_* is such that $\mu_* - L = H - \mu_*$; in other words, $\mu_* = \frac{L+H}{2} = 1050$, and another restriction that emerges is thus that $\sigma < \mu_* - L = H - \mu_* = 650$.

From (2.31), the likelihood for a single observation in this model is

$$l(\mu, \sigma|y_i) = \frac{1}{2\sigma} I(\mu - \sigma < y_i < \mu + \sigma), \tag{2.34}$$

from which after a bit of thought you can see that the complete-

sample likelihood is

$$l(\mu, \sigma|y) = \frac{1}{(2\sigma)^n} I[\mu - \sigma < \min(y)]\, I[\max(y) < \mu + \sigma]. \quad (2.35)$$

Now it is not a lot of fun to multiply (2.32) and (2.35), work out the normalizing constant, and integrate out one of the parameters to get the marginal posterior for the other one, so I am going to use MCMC here. Hastings and Metropolis are certainly possibilities (Problem 2.4), but I thought this would be a good chance to see the third main MCMC technique, **Gibbs sampling**, in action, so let's see how that goes.

Gibbs sampling. The idea behind Gibbs sampling, which (as I mentioned earlier) dates to work by Geman and Geman (1984) in image analysis, is a kind of what-if that is related to the **EM algorithm** (Baum et al., 1970; Dempster, Laird, and Rubin, 1978), a method developed to do maximum likelihood and Bayesian inference in models with missing information. Given a parameter vector $\theta = (\theta_1, \ldots, \theta_k)$ with prior $p(\theta)$, and a sample y with likelihood $l(\theta|y)$, you may well notice that the full posterior $p(\theta|y) = c\, p(\theta)\, l(\theta|y)$ is not so easy to work with, but it would become a lot easier if you only knew the value of some other (missing) information z—in other words, suppose that $p(\theta|y, z)$ is more tractable than $p(\theta|y)$, and could be used to estimate θ (for instance, by taking the posterior mode $\hat{\theta}$ of $p(\theta|y, z)$). Then given an initial estimate $\hat{z} = z_0$, you could construct $p(\theta|y, \hat{z})$, which would give rise to an estimate $\hat{\theta}$, which should lead via $p(z|y, \hat{\theta})$ to a better estimate of z, which would lead via $p(\theta|y, \hat{z})$ to an even better estimate of θ, and so on, around the mulberry bush.

Since marginal posteriors $p(\theta_j|y)$ are of such central interest, a natural way to apply this sort of idea is to let θ_j play the role of θ above and let $\theta_{(j)}$—the θ vector with component j omitted—play the role of z. In the context of the ammonite data, for example, this suggests (1) sampling from $p(\mu|y, \sigma)$, obtaining $\hat{\mu}$ (say), (2) then sampling from $p(\sigma|y, \hat{\mu})$, obtaining $\hat{\sigma}$, (3) then sampling another μ from $p(\mu|y, \hat{\sigma})$, and so on. That, in a nutshell, is Gibbs sampling.

More precisely, for general k the algorithm is summarized by (2.36) below. Demonstrating that the resulting stochastic process $\theta^{(t)}$ is indeed a Markov chain with the right equilibrium[23] distribution and showing how Gibbs fits in with Hastings and Metropolis in the overall MCMC picture (Problem 2.5) are more complicated matters, but if you spot me that it works you can see that the algo-

rithm itself is pretty straightforward: the distributions $p(\theta_j|y,\theta_{(j)})$ are called the **full conditionals** for the model you're sampling from, and the rule is simply that you always use the most recent sampled values of the components of $\theta_{(j)}$ in defining and generating from the next full conditional. One iteration of the repeat loop in (2.36) is called a **scan** of the Gibbs sampler, and fills in one row of the MCMC data set (Table 2.3).

Algorithm (Gibbs sampling) (Geman and Geman, 1984). To construct a Markov chain whose equilibrium distribution is $p(\theta|y)$,

$$
\begin{array}{l}
\texttt{Initialize } \theta^{(0)};\ t \leftarrow 0 \\
\texttt{Repeat } \{ \\
\quad \texttt{Sample } \theta_1^{(t+1)} \sim p[\theta_1|y,(\theta_2^{(t)},\ldots,\theta_k^{(t)})] \\
\quad \texttt{Sample } \theta_2^{(t+1)} \sim p[\theta_2|y,(\theta_1^{(t+1)},\theta_3^{(t)},\ldots,\theta_k^{(t)})] \\
\quad\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\quad \texttt{Sample } \theta_{k-1}^{(t+1)} \sim p[\theta_{k-1}|y,(\theta_1^{(t+1)},\ldots,\theta_{k-2}^{(t+1)},\theta_k^{(t)})] \\
\quad \texttt{Sample } \theta_k^{(t+1)} \sim p[\theta_k|y,(\theta_1^{(t+1)},\ldots,\theta_{k-1}^{(t+1)})] \\
\quad t \rightarrow (t+1) \\
\}
\end{array}
\tag{2.36}
$$

Working out the full conditionals. To apply this idea to the ammonite data, I need to figure out the full conditionals $p(\mu|y,\sigma)$ and $p(\sigma|y,\mu)$. Considering $p(\mu|y,\sigma)$ first, notice that

$$p(\mu|y,\sigma) = \frac{p(\mu,y,\sigma)}{p(y,\sigma)} = c\,p(\mu)\,p(\sigma|\mu)\,l(\mu,\sigma|y). \tag{2.37}$$

The c in (2.37) arises because I don't have to evaluate things like $p(y,\sigma)$ that don't involve μ, since I am thinking of the left side of the equation as a function of μ for fixed y and σ—indeed, anything that appears on the right side that is a function only of y or σ will just get absorbed into the proportionality constant. From (2.32) and (2.35), (2.37) becomes

$$\begin{aligned} p(\mu|y,\sigma) &= \frac{c}{\min(\mu-L,H-\mu)} \cdot I[\sigma < \min(\mu-L,H-\mu)] \cdot \\ &\qquad I[\max(y)-\sigma < \mu < \min(y)+\sigma]. \end{aligned} \tag{2.38}$$

At this point in building your own Gibbs sampler from scratch, you hope that the right side of an expression like (2.38) is the kernel of a density you recognize, so that it will be easy to sample from,

and in fact in many standard situations—for instance, Problems (2.8) and (2.9)—that is how things go. Here I am not so lucky: (2.38) is just a messy bunch of special cases, depending on whether μ is bigger than one thing or smaller than another. I will spare you the details; suffice it to say that, specifying the full conditional for μ in terms of its CDF $F(\mu|y,\sigma)$ instead of its density, you get that

- For $\frac{\max(y)-\min(y)}{2} < \sigma < \mu_* - \min(y)$,

$$F(\mu|y,\sigma) = \left\{ \begin{array}{cc} 0 & \mu < c_1 \\ \frac{\log\left(\frac{\mu-L}{c_1-L}\right)}{\log\left(\frac{c_2-L}{c_1-L}\right)} & c_1 < \mu < c_2 \\ 1 & c_2 < \mu \end{array} \right\}, \tag{2.39}$$

  where $c_1 = \max[L+\sigma, \max(y)-\sigma]$ and $c_2 = \min(y)+\sigma$; and
- For $\mu_* - \min(y) < \sigma < \frac{H-L}{2} \equiv \sigma_*$,

$$F(\mu|y,\sigma) = \left\{ \begin{array}{cc} 0 & \mu < c_1 \\ c_4 \log\left(\frac{\mu-L}{c_1-L}\right) & c_1 < \mu < \mu_* \\ c_4 \log\left[\frac{\sigma_*^2}{(c_1-L)(H-\mu)}\right] & \mu_* < \mu < c_3 \\ 1 & c_3 < \mu \end{array} \right\}, \tag{2.40}$$

  where $c_3 = \min(H-\sigma, c_2)$ and $c_4 = \left\{\log\left[\frac{\sigma_*^2}{(c_1-L)(H-c_3)}\right]\right\}^{-1}$.

The reason I have focused on the CDF rather than the full conditional density for μ is that the next thing I have to do is figure out how to sample from $p(\mu|y,\sigma)$, and one of the easiest ways to do so is to recall (e.g., Ripley, 1987) that $\hat{\mu} = F^{-1}(U|y,\sigma)$ is a draw from $p(\mu|y,\sigma)$ when $U \sim U(0,1)$. So the last step in sampling from μ's full conditional is inverting $F(\mu|y,\sigma)$, which is straightforward: after some algebra you see that to make a draw $\hat{\mu}$ from $p(\mu|y,\sigma)$, you can generate $U \sim U(0,1)$ and

$$\begin{array}{l} \texttt{If } \sigma < \mu_* - \min(y) \texttt{ set } \hat{\mu} = L + (c_1 - L)\left(\frac{c_2-L}{c_1-L}\right)^U; \\ \quad \texttt{else if } U < c_4 \log\left(\frac{\sigma_*}{c_1-L}\right) \\ \quad\quad \texttt{set } \hat{\mu} = L + (c_1 - L)\exp\left(\frac{U}{c_4}\right); \\ \quad \texttt{else set } \hat{\mu} = H - \frac{\sigma_*}{\exp\left[\frac{U}{c_4} - \log\left(\frac{\sigma_*}{c_1-L}\right)\right]} \quad . \end{array} \tag{2.41}$$

The story for σ's full conditional is considerably simpler:

$$\begin{aligned} p(\sigma|y,\mu) &= c\, p(\mu)\, p(\sigma|\mu)\, l(\mu,\sigma|y) \\ &= \frac{c}{\sigma^n} \cdot I[c_5 < \sigma < c_6], \end{aligned} \tag{2.42}$$

where $c_5 = \max[\mu - \min(y), \max(y) - \mu]$ and $c_6 = \min(\mu - L, H - \mu)$. (2.42) can be sampled from in the same way as μ was:

$$F(\sigma|y,\mu) = \left\{ \begin{array}{cc} 0 & \sigma < c_5 \\ \frac{c_5^{1-n} - \sigma^{1-n}}{c_5^{1-n} - c_6^{1-n}} & c_5 < \sigma < c_6 \\ 1 & c_6 < \sigma \end{array} \right\}, \tag{2.43}$$

and to draw a $\hat{\sigma}$ from $p(\sigma|y,\mu)$ you just generate $U \sim U(0,1)$ and

$$\texttt{Set } \hat{\sigma} = \left[(1-U)c_5^{1-n} + Uc_6^{1-n}\right]^{\frac{1}{1-n}}. \tag{2.44}$$

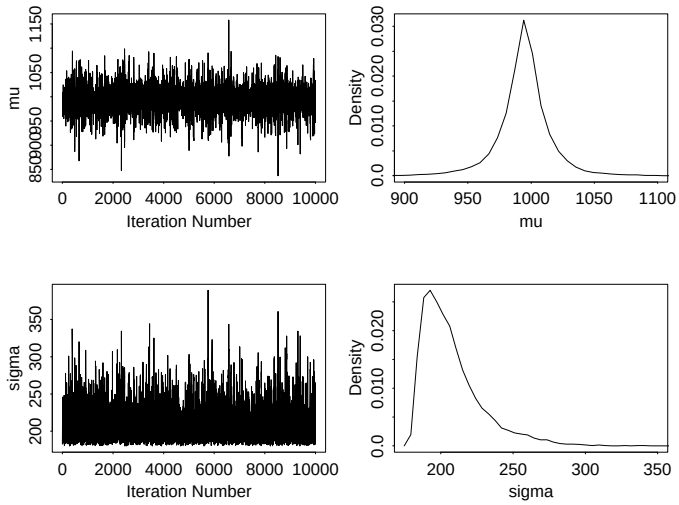

Figure 2.8. *Time series and density traces for μ and σ in the uniform model (2.31, 2.32) applied to the* M. dens. *α data.*

Results for the ammonite data. Section 4 of Appendix 2 contains S+ code implementing this Gibbs sampler in the model (2.31, 2.32), and Figures 2.8 and 2.9 summarize the results when applied to the *M. dens.* α data from Table 2.1. I used a burn-in of 1,000 and a monitoring run (without thinning) of 10,000 from an initial value of $\sigma_0 = 190$ (a bit bigger than the smallest possible value $\frac{1}{2}[\max(y) - \min(y)] = 180$). This took about 2.5 minutes at 333Mhz to produce output that passed all tests in Section 2.4, and resulted (for example) in a Monte Carlo SE for the posterior mean of μ of 0.22. Figure 2.8 shows the time series and density traces for μ and σ (the ACF and PACF plots are not very interesting);

Figure 2.9 repeats for A and B. Table 2.10 contains some numerical summaries for the four parameters.

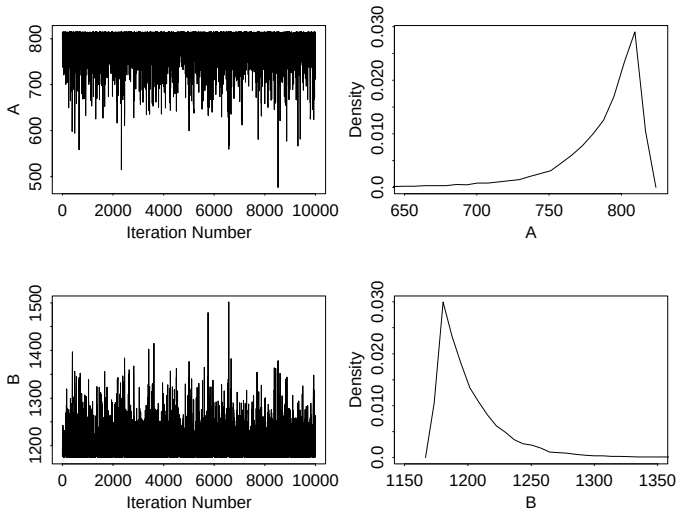

Figure 2.9. *Like Figure 2.8 but summarizing the parameters A and B.*

A number of intriguing things emerge from even a cursory examination of the figures and table.

- The density trace for σ looks more or less like that of a typical scale parameter, and the marginal posteriors for A and B look just about like they would have to, given their status as range-restriction parameters, but the density trace for μ is extremely peaked at its center of symmetry—not at all the Gaussian sort of shape you might expect for a location parameter.
- All four parameters are mixing well, and the first-order autocorrelations $\hat{\rho}_1$ and default $\hat{n}_{RL}$ values are much smaller than those we had come to expect with Hastings or Metropolis (actually, then, what I said a minute ago about the ACF and PACF plots isn't true—they are interesting when compared with those from Hastings or Metropolis precisely because they *don't* show much autocorrelation). This is a general feature of Gibbs—it usually produces parameter chains with less serial correlation (**readers: what's a good reference for this?**).
- The posterior SDs for μ and σ are remarkably small, given the variability of the data values and the small sample size ($n = 16$

for *M. dens.* α): for instance, the posterior SD of μ is only about 0.2% of the posterior mean. By comparison, the sample mean $\bar{y}$ has standard error $\frac{\widehat{SD}}{\sqrt{n}} = 25.3$, so on the variance scale—where frequentists typically compare the performance of estimators—the posterior mean of μ, an alternative estimate of the center of symmetry, is $100\left[\left(\frac{25.3}{20.7}\right)^2 - 1\right] = 50\%$ more efficient than the usual (Gaussian-model-based) sample mean. This is connected to the "witch's hat" shape of the marginal posterior for μ noted above.

Table 2.10. *Numerical summaries of the four parameters in the uniform model (2.31, 2.32) applied to the* M. dens. α *data.*

	Posterior				Default	MLE
	Mean	SD	95% Central Interval	$\hat{\rho}_1$	$\hat{n}_{RL}$	(SE)
μ	994.4	20.7	(948, 1040)	0.044	8800	995 (14.6)
σ	207.9	21.5	(183, 265)	0.54	5000	180 (13.7)
A	786.5	31.0	(700, 814)	0.32	9400	815 (20.0)
B	1202	28.6	(1180, 1280)	0.28	3800	1175 (20.0)

Some insight into what's going on here can be obtained by working out the MLEs of the four parameters in Table 2.10 and their standard errors. It is not hard to show (Problem 2.10) that in model (2.30, 2.31),

$$\begin{aligned}
\hat{A} &= \min(y), \quad E(\hat{A}) = A + \frac{2\sigma}{n+1} \equiv A + \text{bias}, \\
&\qquad V(\hat{A}) = \frac{4\,n\,\sigma^2}{(n+1)^2(n+2)} = V(\hat{B}), \\
\hat{B} &= \max(y), \quad E(\hat{B}) = B - \text{bias}, \\
\hat{\mu} &= \frac{\hat{B}+\hat{A}}{2}, \quad E(\hat{\mu}) = \mu, \quad V(\hat{\mu}) = \frac{2\sigma^2}{(n+1)(n+2)}, \\
\hat{\sigma} &= \frac{\hat{B}-\hat{A}}{2}, \quad E(\hat{\sigma}) = \frac{n-1}{n+1}\sigma, \quad V(\hat{\sigma}) = \frac{n-1}{n+1}\,V(\hat{\mu}),
\end{aligned} \tag{2.45}$$

where the hats denote the MLEs. (I computed the standard errors in Table 2.10 in the usual way, by plugging the MLE for σ into the variance expressions in (2.45).) Since the prior (2.32) I have used for (μ, σ) is quite diffuse, the MLEs should be approximating the marginal posterior modes fairly closely (indeed you can verify

this from Figures 2.8 and 2.9), and the modes and means of these marginals are not all that different, so the variance formulas in (2.45) should provide some guidance as to the uncertainty we have about θ and σ in light of the data.

There is something funny about these variances: they are of order $\frac{1}{n^2}$, instead of the usual $O(\frac{1}{n})$ in location and scale problems. This is due to the extremely light tails of the uniform distribution—in effect, they fall off so rapidly (like a step function, in fact) that you can learn about the range-restriction parameters A and B, and simple functions of them like μ and σ, at a much faster rate than with (say) Gaussian or lognormal data.

Comparing the ammonite taxa. I will finish this case study by applying the methodology developed above to all 13 taxa of ammonites in Tables 2.1 and 2.2. I used the same model and sampling strategy as those that produced Table 2.10 to generate 5,000 draws from each of the 13 posterior distributions for μ, one for each taxon, and the results are summarized in Figures 2.10 and 2.11.

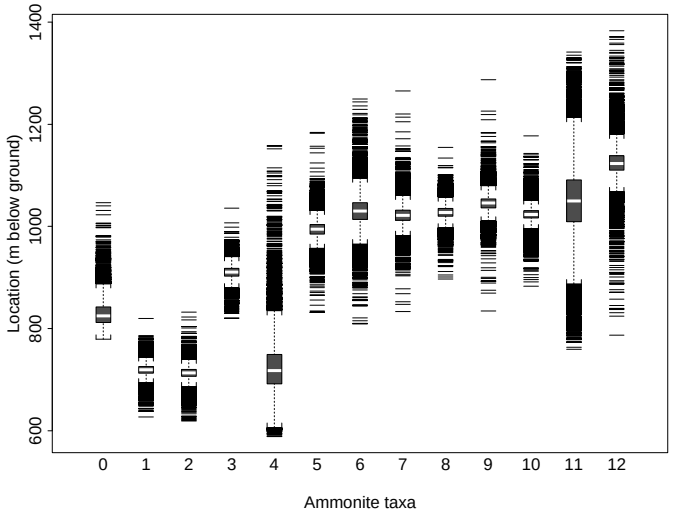

Figure 2.10. *Parallel boxplots of the 5,000 draws from the posteriors for μ in the ammonite uniform example, contrasting the 13 taxa in Tables 2.1 and 2.2.*

A natural way to examine these 13 posteriors is with parallel boxplots, as in Figure 2.10—I have used the same numbering scheme as Strauss and Sadler, who (for some reason) ranked the taxa in

increasing order of their smallest observations. Most of the distributions in this set of boxplots are close to symmetric, as you would expect from the model (in fact, sharp lack of symmetry in the posterior for μ could be a model diagnostic here). It is also interesting to note that some of the most unusual posteriors arise from the tiniest samples (taxa 4 and 11 each had a sample size of only 3, for instance).

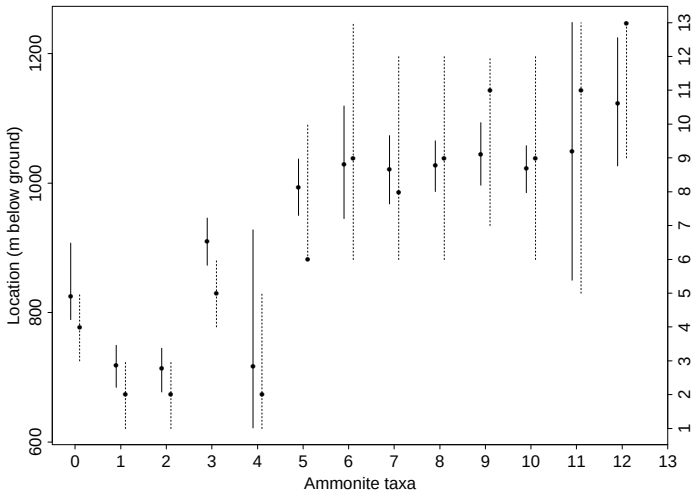

Figure 2.11. *Plots of the posteriors for μ and the ranks of the μ's in the full ammonite data set. The vertical lines plot 95% central intervals and the superimposed points are medians in all cases. The left-hand (solid) line for each taxon is the posterior for μ; the right-hand (dotted) line summarizes the posterior for the ranks.*

One scientific question arising from the collection of the data in Tables 2.1 and 2.2 was the order in which the ammonite taxa secured their evolutionary niches in the Cretaceous, for instance as measured by the center μ of their true ranges. Figure 2.11 was created to help answer this question, and requires a bit of explaining. There are two vertical lines plotted for each taxon: on the left in each case (and referring to the left-hand vertical scale) is the 95% central interval for μ (with the median of the posterior superimposed as a dot); the right (dotted) line (with reference to the right-hand vertical scale) gives the 95% central interval for the *ranks* of the μ's.

How did I get the intervals for the ranks? As noted by Spiegelhalter et al. (1995), whose rank analysis of an entirely different kind of data motivated Figure 2.11, an excellent feature of MCMC is the ease with which unusual and complicated functions of the underlying parameters may be monitored and summarized alongside the quantities appearing in your models. In this case, since I am treating the μ's for the various taxa as entirely independent in this modeling, all I had to do to monitor the ranks was (a) to make an MCMC data set with 5,000 rows and 13 columns, one for each of the sets of posterior draws for the different taxon μ's, and (b) create a new derived data set in which each row is replaced by the ranks of the observations in that row.

Two interesting things are immediately apparent from this plot:

- Uncertainty about the ranks of the μ's is considerably larger than that about the μ's themselves—notice in particular the disparity between the vertical line lengths for taxa 6–10. With some assurance we can say that the range-centers for taxa 0–4 came earlier in the Cretaceous than those for taxa 5–12, but that's about all we're pretty sure of. This contrasts with the rather sharper ranking conclusions that appear possible from looking directly at the posteriors for the μ's; and
- If you examine the dots in Figure 2.11 you will notice a pattern: for both the lowest taxa, 0–5, and the highest ones, 8–12, the rank median is farther from the center than the μ median. Starting in Chapter 3 we will see examples of what are called **shrinkage** estimates, in which extreme values, in comparisons like those in this figure, are pulled back in toward the middle by switching over to a different model for the data. But because *something* has to get rank 1 even if it is only a little bit smaller than the second-smallest thing (and analogously for the upper end of the scale), switching attention from the underlying μ's to their underlying ranks evidently produces a set of *anti-shrinkage* or *expansion* estimates.

2.6 Case study: Measurement of physical constants

Back at the end of Chapter 1, when we were looking at the NB10 data (Case Study 1.2), you may remember that the Gaussian model of Section 1.8 didn't fit very well, because of a number of outliers in both tails (see Figures 1.2 and 1.4). At the time I said that it would

perhaps be good to *expand* the Gaussian model, by embedding it in the t family and adding a parameter ν for tail-weight. We couldn't fit that model in Chapter 1 because conjugacy is not available when ν is treated as an unknown, but MCMC makes fitting models like this pretty close to routine, as I will now try to show.

In the notation of Chapter 1 the expanded model is

$$\begin{aligned} (\mu, \sigma, \nu) &\sim p(\mu, \sigma, \nu) \\ (y_i | \mu, \sigma, \nu) &\overset{\text{IID}}{\sim} t_\nu(\mu, \sigma^2), \quad i = 1, \ldots, n. \end{aligned} \tag{2.46}$$

This model is actually not very hard to fit with Gibbs sampling—I will conclude the chapter with an illustration of this using an MCMC package called `BUGS`—but I thought I would take the opportunity first to use (2.46) to lay out a fairly generic strategy for Bayesian model-fitting based on Metropolis sampling.

This is our first MCMC example in which the parameter vector θ has dimension k bigger than 1, and the first thing you might think of in trying to apply Metropolis is to propose $N(0, \kappa_j \sigma_j^2)$ moves, like I did in Section 2.3, separately and independently for each parameter θ_j. In other words—in model (2.46), for example—in filling in each row of the MCMC data set you might first sample a μ, then a σ, and then a ν (rather like Gibbs sampling, except that the draws in this case would be independent of all other values in the MCMC data set). This would be fairly easy to code, but it has a big potential flaw: if the parameters are highly correlated, then a lot of the moves you propose by treating them as independent will be implausible, and your acceptance rate will be far from optimal.

A generic Metropolis sampling strategy. To improve on this, the idea behind the generic strategy I want to look at here is that, possibly after appropriate reparameterization, the posterior distribution for θ should be close to *multivariate* normal for moderate to large n, say $p(\theta|y) \dot{\sim} N(\theta^*, \Sigma)$. This suggests a kind of generalization of the idea in Section 2.3: a random-walk Metropolis with multivariate normal proposal distribution, centered at where you are now, and with covariance matrix a multiple κ of Σ, for suitably chosen κ. This will accurately reflect any posterior correlations, thereby improving the efficiency of the sampling.

The steps of the strategy are thus as follows.

(1) Transform any components of θ that live only on a subset of the real line to all of $\mathbb{R}$. Rewrite the log likelihood in this new

parameterization, and recompute the log prior by including the appropriate Jacobian.

(2) Use pencil and paper or (more reliably, in complicated problems) a symbolic computing package to find the posterior mode θ_m. Symbolically obtain the Hessian H (the second partial derivative matrix) of the log posterior, evaluate it numerically at the posterior mode, and compute $\hat{\Sigma} = -H^{-1}\big|_{\theta_m}$. If the prior is diffuse you can replace "posterior mode" by "MLE" and "log posterior" by "log likelihood."

(3) Code up and run a Metropolis sampler that makes $N(0, \kappa\hat{\Sigma})$ moves, varying κ to minimize the maximum of the SSIF or $\hat{n}_{RL}$ values across the components of θ. Gelman et al. (1996) have shown that, in a particular class of problems that should give some guidance here, the optimal κ behaves roughly like $\frac{5.8}{p}$, and the optimal acceptance rate decreases from about 0.44 for $p = 1$ to about 0.27 for $p = 10$, roughly along the curve $0.23 + \frac{0.26}{p} - \frac{0.046}{p^2}$.

If step (2) is too difficult, you will need another way to get an approximate Σ. The simplest idea is probably to use the independent-component sampler I criticized a few paragraphs ago to get yourself started, and then switch over to step (3). One nice thing about MCMC is that, even with an inefficient proposal distribution, the output of the chain—once equilibrium has been reached—is a valid sample from the posterior. So you can try an iterative strategy like the following: start with a poorly-tuned proposal; run it a very long time; use the sample covariance matrix based on the columns of the resulting MCMC data set as an initial estimate $\hat{\Sigma}_0$ of Σ; run for a long time with a multivariate normal proposal based on $\hat{\Sigma}_0$; use the sample covariance matrix from the columns of *this* MCMC data set to produce a better estimate $\hat{\Sigma}_1$; and so on.

This is called **adaptive** Metropolis(-Hastings) sampling (e.g., Gilks et al., 1997), and there is only one thing to watch out for: if you keep indefinitely refining the proposal distribution adaptively, based on the previous output of the chain, it has been shown that the sampler will not (necessarily) converge to the right equilibrium distribution. So you need to stop the adaptive process at some point before monitoring to produce the results you will announce, because if not you may well be monitoring the wrong distribution.

In a bit more detail, the alternative strategy is as follows.

(2′)(a) Code up a Metropolis sampler that makes a series of $N(0, \kappa_j\sigma_j^2)$

moves, one for each parameter θ_j, obtaining estimates for the σ_j^2 by whatever means you can think of (likelihood theory, iterative guesswork, ...) and varying the κ_j so that the product of the acceptance probabilities is as large as you can make it, up to a maximum of about 0.5. Set $s = 0$.

(b) Run this sampler for a long time from a good starting value and with a bigger-than-usual burn-in (use SSIF and/or $\hat{n}_{RL}$ values to define "large"), and use the sample covariances of the columns of the resulting MCMC data set to construct an estimate $\hat{\Sigma}_s$ of Σ. If $s > 0$ and $\hat{\Sigma}_s$ and $\hat{\Sigma}_{s-1}$ don't differ too much, go to (4′).

(3′) Code up and run a Metropolis sampler that makes $N(0, \kappa\hat{\Sigma}_s)$ moves, varying κ to optimize the acceptance probability as usual. When you have a κ you like, increment s and go back to (2′b).

(4′) Now, finally, make your monitoring run for the money using the most recent $\hat{\Sigma}_s$.

I have implemented strategy (1–3), using `S+` and the symbolic computing package `Maple`, with model (2.46) using the NB10 data (Problem 2.11 invites you to try strategy (1, 2′–4′) on this same example). This requires creating the new parameters $\eta = \log(\sigma)$ and $\lambda = \log(\nu)$, and rewriting the model in terms of $\theta = (\mu, \eta, \lambda)$. The log likelihood function, in this parameterization, is (from Appendix 1)

$$\begin{aligned} \log[l(\mu,\eta,\lambda)|y] &= c + n\log\left[\Gamma\left(\frac{e^\lambda+1}{2}\right)\right] - n\eta \\ &\quad -n\log\left[\Gamma\left(\frac{e^\lambda}{2}\right)\right] - \frac{n\lambda}{2} \\ &\quad -\frac{e^\lambda+1}{2}\sum_{i=1}^{n}\log\left[1+e^{-(\lambda+2\eta)}(y_i-\mu)^2\right], \end{aligned} \tag{2.47}$$

where $\Gamma(\cdot)$ is Euler's gamma (generalized factorial) function (e.g., Abramowitz and Stegun, 1972).

Prior elicitation in the NB10 t model. By way of a prior I have tried to bring in a modest amount of information that accords with the science of the problem. With $n = 100$ observations it should be OK to use a prior with independent components, because any strong posterior correlations that should be present will be accurately learned from the data, and having transformed to

the log scale for σ and ν it should be reasonable to work with Gaussians, so I took

$$p(\mu, \eta, \lambda) = N(\mu|\mu_0, \sigma_\mu^2)\, N(\eta|\eta_0, \sigma_\eta^2)\, N(\lambda|\lambda_0, \sigma_\lambda^2). \tag{2.48}$$

This reduces the elicitation problem to that of specifying the prior means and SDs for each of μ, η, and λ.

- μ represents the true weight of the block of metal NB10, which is supposed to weight around 10g, and the observations are in micrograms below this nominal weight. So to give the National Bureau of Standards (NBS) the benefit of the doubt I should probably take $\mu_0 = 0$, but σ_μ should be big to reflect the possibility of substantial bias on the microgram scale. Based on previous results with similar weighing equipment at the NBS (Ku, 1969), I have chosen $\sigma_\mu = 500$ in what follows (see Problem 2.12 for a sensitivity analysis of the prior specification in this case study).
- ν indexes the tail-weight of the true distribution of measurement errors. Churchill Eisenhart (1979, personal communication), a leading statistician at NBS for decades, is on record as saying that "Measurement error processes in the physical sciences, when investigators report all their apparently valid data, tend to behave roughly like t on about 7 degrees of freedom." I have interpreted this expert judgment, a bit liberally, as an approximate statement that $P(2 < \nu < 20) = 0.95$. On the log scale this creates a 95% prior interval for λ of $(0.69, 3.0)$, which in the Gaussian world implies $\lambda_0 = 1.84$ and $\sigma_\lambda = 0.59$.
- σ is related to the true SD of the measurement errors made by the NBS weighing process, through $SD(y) = \sqrt{\frac{\nu}{\nu-2}}\,\sigma$ (as long as $\nu > 2$). A conservative reading of Ku (1969) suggests that errors on the order of 1–200 on the microgram scale are possible, which I will translate into the statement $P(1 < \sqrt{\frac{\nu}{\nu-2}}\,\sigma < 200) = 0.95$ for elicitation purposes. Taking $\nu \doteq 7$ for simplicity, for $\eta = \log(\sigma)$ this statement corresponds to the 95% prior interval $(-0.17, 5.13)$, leading to a Gaussian prior mean of $\eta_0 = 2.48$ and SD of $\sigma_\eta = 1.35$.

Before doing any sampling it is worth looking at the log posterior a bit to see if any pathologies should be anticipated. I can't plot $p(\mu, \eta, \lambda|y)$ in all its glory, because we are condemned to three visual dimensions, but—like the blind men and the elephant in the

old story—I can try to create a mental image of the whole thing by looking at various views of it one by one. Figure 2.12 presents four such views of the log posterior in this problem. I drew the upper left panel by holding constant μ and η at plausible values, not (perhaps) too far from their posterior modes (I took the sample mean for μ and the log of the sample SD for η), and tracing out the log posterior as a function of λ. If the posterior is multivariate normal this plot should look locally quadratic around its maximum, and—while it lacks a bit in the symmetry department—it is at least bowl-shaped down with only a single maximum. The other two similar plots (not shown), obtained by fixing (μ, λ) and (η, λ), are also reasonably well-behaved—in particular, there are no signs of multimodality.

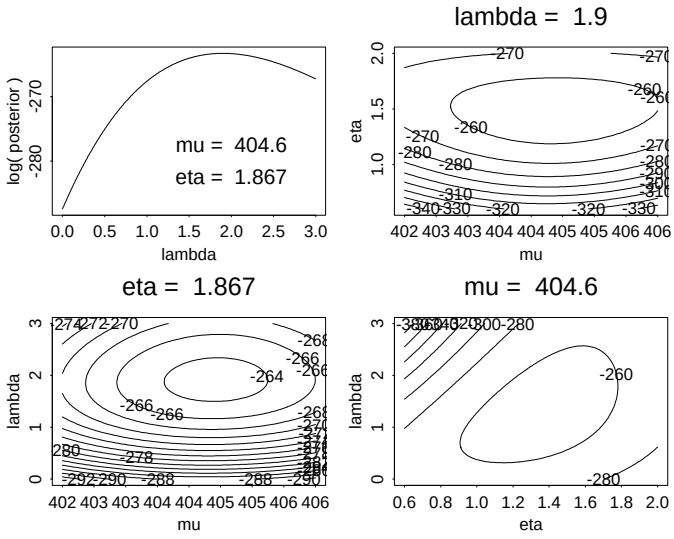

Figure 2.12. *Four exploratory views of the log posterior in the NB10 t model. The upper left panel is a log posterior trace varying λ for fixed values of μ and η; the other three panels are contour plots of the log posterior, fixing one parameter in each case and letting the other two vary.*

The other three panels in Figure 2.12 are contour plots of the log posterior, obtained by fixing one component of θ at a time and letting the other two vary. Each of these graphs should look like a set of concentric ellipses if the posterior is close to multivariate normal, and as long as you cast a slightly generous eye on the (η, λ)

plot you can see that things are not *terribly* far from MVN. It's also interesting to note that η and λ are fairly strongly positively correlated in the posterior, which on reflection makes sense: if I gave you a moderate-n sample of data with a few points that may or may not be outliers, it would be hard for you to tell if the underlying story was (small ν, small σ)—in other words, the data really are t_ν—or (large ν, large σ), which is like saying that the data are really (close to) Gaussian but just with a big SD. Thus scale and shape are confounded in the t family.

OK, now I'm ready to try to build my Metropolis sampler. I've done step (1) already (I didn't have to compute any Jacobians because it was easy to elicit on the transformed scale). Appendix 2 Section 5 gives some code in the symbolic computing language `Maple` to get the approximate posterior covariance matrix $\hat{\Sigma}$. It's easy to specify the log posterior in `Maple` since it has a built-in $\log[\Gamma(\cdot)]$ function, and then you just have to ask `Maple` to differentiate the log posterior symbolically and solve the resulting **MAP** (maximum *a posteriori*) equations numerically to find the mode.

It turns out that `Maple` could not find the mode θ_m without some help, in the form of range restrictions on where θ_m might be, but after I gave it a rather broad hint of this kind it was able to solve the likelihood equations in this problem with no trouble, in about 2.5 seconds at 333Mhz. To finish the calculation off I just have to ask `Maple` to calculate the Hessian H symbolically and evaluate it numerically at the mode, and then take $\hat{\Sigma} = -H^{-1}\big|_{\theta_m}$, as noted in the generic strategy above. The results are $\theta_m = (404.3, 1.346, 1.260)$ and

$$\hat{\Sigma} = \begin{pmatrix} 0.215 & 0.00299 & 0.00808 \\ 0.00299 & 0.0119 & 0.0149 \\ 0.00808 & 0.0149 & 0.0749 \end{pmatrix}, \tag{2.49}$$

leading to approximate standard errors $(\sqrt{0.216} = 0.46, 0.11, 0.27)$ for $(\hat{\mu}, \hat{\eta}, \hat{\lambda})$ and approximate correlations $(\frac{0.00299}{\sqrt{0.215 \cdot 0.0119}} = 0.059,$ $0.064, 0.50)$ for $[(\hat{\mu}, \hat{\eta}), (\hat{\mu}, \hat{\lambda}), (\hat{\eta}, \hat{\lambda})]$. Thus μ is around 404.3 ± 0.46; η is about 1.346 ± 0.11, meaning that σ is likely to be in the range $(\exp(1.346 - 2 \cdot 0.11) = 3.1, \exp(1.346 + 2 \cdot 0.11) = 4.8)$; and λ is around 1.260 ± 0.27, so that ν is probably in the interval $(2.1, 6.0)$. All of this is useful information in extracting the full posterior.

Appendix 2 Section 6 contains some `S+` functions to implement the generic Metropolis strategy above with the NB10 data (note

how little would have to be changed to use this sampler on a completely different problem). With $\kappa = 2 \doteq \frac{5.8}{3}$, a single long run of 45,000, storing every 9th iterate, after a burn-in of 2000 from a starting value of $\theta_0 = (404.6, 1.699, 1.946) = [\bar{y}, \log(\sqrt{\frac{5}{7}}\, s), \log(7)]$ took about 10 minutes at 333Mhz and produced the output summarized in Table 2.11 and Figure 2.13. All `CODA` diagnostics were well-behaved, and the thinning by a factor of 6 resulted in fairly low serial correlations and default $\hat{n}_{RL}$ values. The Metropolis acceptance rate was 0.31, which is near-optimal for $p = 3$ based on Gelman et al. (1996)'s results, so I didn't try to look for a better κ.

Table 2.11. *Numerical summaries of the three original-scale parameters in the t model (2.46, 2.48) applied to the NB10 data, using the monitoring strategy described in the text.*

	Posterior					
	Mean	SD	95% Central Interval	$\hat{\rho}_1$	Default $\hat{n}_{RL}$	MLE (SE)
μ	404.3	0.48	(403.4, 405.3)	0.20	5000	404.3 (0.46)
σ	3.92	0.44	(3.14, 4.87)	0.17	4100	3.70 (0.42)
ν	3.75	1.1	(2.15, 6.44)	0.16	3900	3.01 (0.86)

The table and figure bring up several interesting points.

- The posterior mean of the scale parameter σ is substantially lower than the sample SD $s = 6.5$, but this is to be expected since $V(y) = \frac{\nu}{\nu-2}\sigma^2$ in this model (as long as $\nu > 2$). Indeed, the sample average of the quantity $\frac{\nu_* \sigma_*^2}{\nu_* - 2}$ across the 4,971 rows of the MCMC data set with $\nu_* > 2$ is 44.4, not far from the sample variance 41.8.

- Bayesian and ML inferences with these data (with my prior, at least) are similar for μ, but the posterior means are about 6% and 25% larger than the MLEs for σ and ν, respectively, and the MLE standard errors are smaller than the posterior SDs. Part of this difference comes from the prior, part from the difference between means and modes for skewed distributions, and part from the way ML inference (sometimes inaccurately) deals with uncertainty about $\theta_{(j)}$ when summarizing uncertainty about θ_j. Note in particular how much smaller the SE for $\hat{\nu}_{\text{MLE}}$ is than the posterior SD for ν.

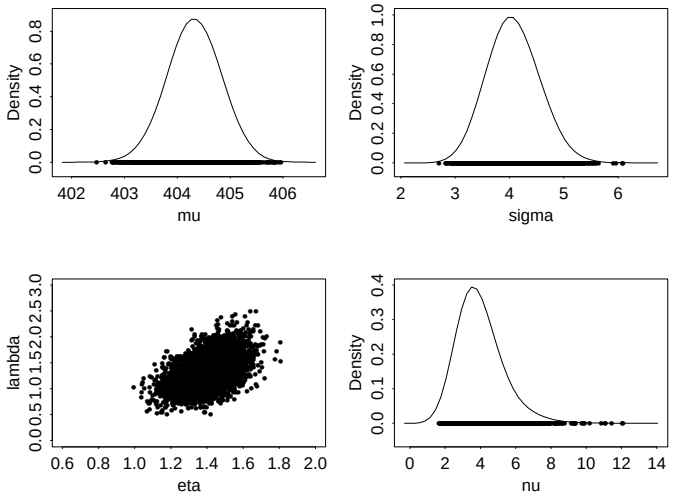

Figure 2.13. *MCMC output summaries in the NB10 t model: density traces of μ, σ, and ν (reading clockwise from the upper left), and scatterplot of λ versus η (compare with the lower right contour plot in Figure 2.12).*

- Most importantly, recall from Chapter 1 that when we (incorrectly) assumed a Gaussian model for these data, the posterior mean and SD of μ (which has the same meaning in both models, and is thus comparable) were 404.3 and 0.65. With the t model the posterior mean is the same but the posterior SD, 0.47, is substantially (28%) smaller.
  - This makes sense from a frequentist robustness point of view: if—in view of the outliers—you were to use a trimmed mean instead of the sample mean, you would lop off the smallest and largest (say) 2 observations, calculate the mean $\bar{y}_T$ (also about 404.3) and SD s_T (considerably smaller: 4.25) of the rest, and (in effect) use $\frac{s_T}{\sqrt{96}} = 0.43$ as your standard error.
  - However, the conclusion is interesting from a Bayesian robustness point of view: when I expand the Gaussian model by embedding it in the t_ν family for unknown ν, my *model uncertainty* has increased (because the former model is a special case of the latter, obtained by pretending you know that $\nu = \infty$), but evidently in this case my *inferential* uncertainty

about the quantity of principal interest—μ—has *decreased*. This is partly because the t_ν model with small ν fits better and partly because it turns out that the Gaussian is a very conservative choice for inference about location parameters (in fact, it minimizes Fisher information for such parameters in [essentially] the [whole] class of symmetric distributions; see Draper, 1997).

One last point to consider before leaving this case study concerns **model diagnostics**, a topic I will take up in more detail in Chapter 4. I have been reasonably careful about *MCMC* diagnostics in this chapter, but it is all too easy in the midst of looking at `CODA` output to forget that (a) *model* diagnostics are equally important and (b) MCMC diagnostics have little or nothing to say directly about the fit of the model to the data.

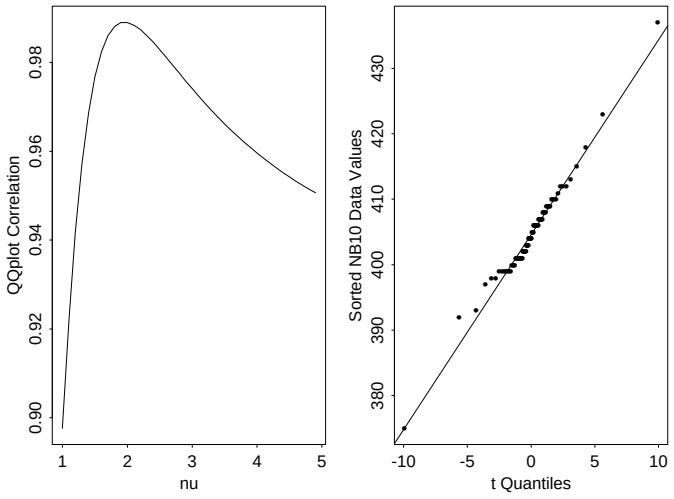

Figure 2.14. *Simple model diagnostic plots in the NB10 example. The left panel relates the correlation of the t_ν quantiles and the sorted NB10 data values to ν, and the right panel is a t_ν qqplot of the NB10 data with $\nu = 2$.*

Figures 1.2 and 1.5 demonstrated that the Gaussian model fit the NB10 data poorly, but did not directly show that the t model fits well. Figure 2.14, on the other hand, provides some evidence that the t family is appropriate for the NB10 data. With access to the CDF of the t_ν distribution it is easy enough to make a t

qqplot of a data set, but what should you choose for ν? The left panel of Figure 2.14 gives one simple answer: I have plotted the correlation between the t_ν quantiles and the sorted NB10 data values[24] as a function of ν, and you can see that this is maximized for $\nu \doteq 2$. So the right panel of the figure gives a t_2 qqplot, which does indeed fit the data pretty much like a glove. It is interesting to consider (Problem 2.13) why the apparently optimal value of ν in this sense is not particularly well supported by the posterior for ν summarized in Table 2.11.

A Gibbs shortcut: `BUGS`. The generic Metropolis `S+` analysis above (or—better—its analogue in `C`) is a reasonably satisfying way to implement MCMC in many problems, but it would be nice if there were a rather more user-friendly environment in which to get MCMC results. There have been several attempts to date to supply fairly narrowly-targeted MCMC packages, including `bpois` (for Poisson regression; Doss and Narasimhan, 1994) and `MCSim` (by Frédéric Bois; see Carlin and Louis, 1996); the most successful general-purpose attempt so far—by quite a margin—has been the Gibbs sampling package `BUGS`, developed by David Spiegelhalter, Wally Gilks, and colleagues at the MRC Biostatistics Unit in Cambridge (UK). The program is available for free, in a variety of hardware and operating system configurations, at `http://www.mrc-bsu.cam.ac.uk` or by anonymous ftp at `ftp.mrc-bsu.cam.ac.uk`, and may be run either in interactive or batch mode. The authors have provided excellent documentation for their code, including an extensive set of worked examples.

At first thought, writing a generic Gibbs sampling package sounds like a daunting task—for instance, how would you automatically figure out the full conditional distributions for an arbitrarily specified model? The authors of `BUGS` (Gilks et al., 1994) have succeeded in achieving considerable generality by means of two fairly mild forms of limitation, as follows.

- With a few exceptions, `BUGS` is restricted to fitting models expressible as **directed acyclic graphs** (DAGs) (e.g., Whittaker, 1990; Lauritzen et al., 1990). Figure 2.15 presents a visual representation of the DAG implied by the NB10 t model (2.46, 2.48). In pictures such as these, *constants* (not present here) are denoted by rectangles; *stochastic nodes* are variables given a distribution by the model, and are denoted by circles; and *directed links* between nodes are indicated by arrows: solid arrows denote

stochastic dependence, and dashed arrows (not present here) indicate deterministic relationships. The directed links basically specify what depends on what in the model: the node into which an arrow points is dependent on the node from which the arrow came.

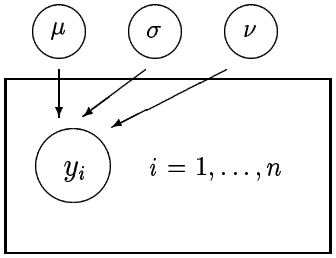

Figure 2.15. *DAG representation of the variables in the NB10 t model* (2.46, 2.48).

The term *directed* in the acronym DAG means that all the lines connecting nodes in the graph have to have arrows (so that you can figure out the dependence structure); and *acyclic* means that there are no subsets of the graph in which you can start somewhere, follow the arrows along, and end up back where you started. The DAG in Figure 2.15 is incredibly simple (see the examples manuals that come with `BUGS` for illustrations of how complicated things can get)—about all you can learn from this DAG is that the y_i depend on μ, σ, and ν (in graphical models jargon, the three latter nodes are *parents* of the y_i, or equivalently y_i is a *descendant* (or *child*) of each of the three parameters), and that μ, σ, and ν are independent of each other (in the prior specification).

The reason `BUGS` sticks to DAGs (and this is essentially a null restriction in this book—I cannot think of any model we will look at that is not a DAG) is that it is (relatively) easy to specify the full conditionals with DAGs, at least up to multiplicative constants (which are irrelevant in Gibbs sampling): the full conditional for a given stochastic node is just (proportional to) the product of the distribution specified at that node and the distributions specified at all of its children. Thus the tree structure implied by the DAG (and the acyclic assumption ensures that it is indeed a tree, and not something more complicated) allows the full conditionals to be read off directly.

- In addition to a general way to figure out the full conditionals, BUGS needs a strategy for sampling from them. The developers have adopted a three-part approach to dealing with this, as follows. First, the code contains a simple expert system that tries to recognize conjugacy and make use of standard methods of sampling from conjugate distributions, if this avenue is successful. Next, if this fails, BUGS tries to convince itself that the full conditional distributions of the model are *log-concave*, which just means that on the log scale these distributions should be bowl-shaped down[25]. The reason for this restriction is that Gilks and Wild (1992) and Gilks (1992) have developed a clever *adaptive-rejection* sampling method that relies on the log-concavity to create a progressively more efficient rejection sampler over time as the sampling proceeds, and BUGS uses this approach when conjugacy fails. Finally, if neither conjugacy nor log-concavity is available, BUGS can sample from an arbitrary full conditional if it is *discretized*. More support points in the discrete approximation to the real full conditional will obviously increase the accuracy of the approximation, but too many such points will produce a very slow sampler indeed.

I will conclude this chapter with an alternative analysis of the NB10 data using Gibbs sampling in BUGS. Table 2.12 gives the principal input file for the BUGS reanalysis, which is given a name with suffix `.bug` in the directory where you want to run BUGS. You can see that the syntax is clear and resembles that of S+ in some respects (to aid users of the former who are already familiar with the latter). You name your model; specify the values of any constants; tell BUGS which of the names you will be using in the program should be thought of as variables, and with what (vector or matrix) dimensions; tell it the names of the files from which it can read in the data (suffix `.dat`) and the initial values (suffix `.in`) for the MCMC sampling; specify the priors; specify the likelihood; and define any derived quantities for monitoring.

Appendix 2 Section 7 contains the other files used to make BUGS runs in this problem. After writing the `.bug` file and making sure that the `.dat` and `.in` files are correctly specified, you run BUGS either interactively or in batch mode. In the former case you type `bugs` and enter commands from the keyboard; in the latter you collect all of these commands into a file with suffix `.cmd` and type (for instance) `backbugs nb10.3.cmd`. The main advantages of batch

mode are that you don't have to wait around for a really long burn-in to finish, to type in commands that govern the monitoring phase, and that you can use it in writing simulations based on BUGS.

Table 2.12. BUGS *file* nb10.3.bug *for a Gibbs sampling reanalysis of the NB10 data.*

```
model nb10.3;                    # Naming the model.

const
  n = 100, g = 101;              # Defining the constants.

var
  mu, tau, u, grid[g],           # Specifying the variables.
    nu, y[n], sigma;

data y in "nb10-y.dat",          # Reading in the data
  grid in "nb10-grid.dat";       #   and initial values.
inits in "nb10.3.in";            #

{
  mu ~ dnorm( 0.0, 4.0E-6 );     # Specifying the priors
  tau ~ dgamma( 0.25, 0.12 );    #   for mu, sigma, and nu
  u ~ dcat( grid[] );            #   (see text).
  nu <- 1.0 + u / 7.0;           #

  for ( i in 1:n ) {             #
    y[i] ~ dt( mu, tau, nu );    # Specifying the likelihood.
  }                              #

  sigma <- 1.0 / sqrt( tau );    # Defining a derived quantity.
}
```

Either way—interactive or batch—your first command will be, for example, `compile "nb10.3.bug"`, after which BUGS will tell you about all your syntax errors and quit if it finds any. It often takes several iterations of editing the `.bug` file before you have a clean compile, which is why most people run BUGS interactively until they have (ahem) gotten all the bugs out before going to batch mode. After the compilation, your next command is usually something like `update(1000)`, which will (in this case) perform a burn-in of 1,000 iterations; after that you prepare for the monitoring phase by issuing a series of commands that tell BUGS what, and how, to monitor. I have used `monitor(mu,14)`, `monitor(sigma,14)`, and `monitor(nu,14)` followed by `update(70000)` in Appendix 2 Section 7—this has the effect of requiring BUGS to store monitored values for μ, σ, and ν (without doing so for any other variables

in Table 2.12) but with a thinning ratio of 14 across the 70,000 iterations, so that only $\frac{70000}{14} = 5{,}000$ values are actually written to disk for each variable. The command `q()` makes BUGS actually do the writing out to disk, after which it tidies up a bit and quits.

Model specification in BUGS. Specifying the likelihood in Table 2.12 is pretty straightforward: BUGS is able to work with something like 21 different built-in distributions that routinely arise in Bayesian analysis, of which $t_\nu(\mu, \sigma^2)$ is one. Variances (and related scale parameters) in BUGS are always specified by working directly instead with precisions; the quantity τ in the BUGS statement `y[i] ~ dt( mu, tau, nu );` is just given by $\tau = \frac{1}{\sigma^2}$. This explains why I have to define the derived quantity `sigma <- 1.0 / sqrt( tau );`—I would rather monitor σ than τ. The `for ( i in 1:n )` loop is just telling BUGS in its language that, conditional on (μ, τ, ν), the n observed NB10 measurements y_i are IID $t_\nu(\mu, \sigma^2)$.

In making this BUGS run I wanted to more or less duplicate the earlier Metropolis analysis of this problem, and it turned out that specifying the prior equivalently in BUGS required a bit more work. First I tried a literal translation of the independent normal and lognormal priors for μ, σ, and ν used previously:

```
mu ~ dnorm( 0.0, 4.0E-6 );
sigma ~ dlnorm( 2.48, 0.549 );
nu ~ dlnorm( 1.84, 2.87 );
tau <- 1.0 / pow( sigma, 2 );
```

Here `dnorm( 0.0, 4.0E-6 )` means a normal distribution with mean 0 and precision 0.000004, which corresponds directly to the earlier specification $\sigma_\mu = 500$, and the parameters of the lognormals are the same as those I used earlier (with the variability again on the precision scale—for example, $\sigma_\lambda = 0.59$ earlier, which translates into a prior precision for $\nu = e^\lambda$ of $\frac{1}{0.59^2} = 2.87$). However, at the end of the compile phase with this prior BUGS announced

```
Error in file: nb10.1.bug
 for node: sigma
 -- error  --
Unable to choose update method for node
```

which is its way of saying it cannot verify that the full conditional for σ is log concave[26] with this model, and that therefore it doesn't know how to sample.

So I said to myself, OK, given that BUGS likes to work with variance parameters on the precision scale and the conjugate prior

for precisions in Gaussian models is the gamma distribution (this follows from the conjugate prior for the variance being the scaled inverse χ^2), it probably doesn't like the lognormal prior for σ, because the induced prior on τ is not gamma. So next I tried

```
mu ~ dnorm( 0.0, 4.0E-6 );
tau ~ dgamma( 0.001, 0.001 );
nu ~ dlnorm( 1.84, 2.87 );
```

starting (initially) with an extremely diffuse prior for τ (the $\Gamma(\epsilon, \epsilon)$ prior for precisions, for small ϵ like 0.001, has a big spike near 0 but is close to flat over the entire rest of the real line; see Spiegelhalter et al., 1995). However this time BUGS said

```
Error in file: nb10.2.bug
 for node: nu
 -- error  --
Unable to choose update method for node
```

meaning that the prior for σ (through τ) was OK, but now it was having the same log-concavity[26] trouble with ν.

So conjugacy and log-concavity (appear to) fail for ν, leaving the discretization approach as the only way to work with parameters like this in BUGS. The code will allow you to work with arbitrary discrete distributions with support points $\{1, \ldots, K\}$, for $K \leq 500$ (which is plenty to get good accuracy), and you can then transform the 1–K scale linearly to any other finite range. To approximate the prior I used in the generic Metropolis approach, I want to create a discretized version of the lognormal distribution with mean and SD (on the log scale) $\lambda_0 = 1.84$ and $\sigma_\lambda = 0.59$, except truncated to an interval (l, h) wide enough to include the entire likely posterior for ν. To achieve this I (i) chose $l = 1.1, h = 15.4$, and $K = 101$; (ii) got S+ to work out the mass at each point based on its lognormal CDF; and (iii) stored these 101 numbers in the file `nb10-grid.dat`. Then the two statements (a) `u ~ dcat( grid[] )` and (b) `nu <- 1.0 + u / 7.0` in the `.bug` file act (a) to create a random draw u from the discretized distribution spread out from 1 to 101 and (b) to transform this distribution to live on (1.1, 15.4), as desired.

The last thing to specify is the hyperparameters of the gamma prior on τ. I did this by reasoning that if $\sigma \sim LN(2.48, 1.35^2)$ then $\sigma^2 \sim LN(4.96, 2.70^2)$ and then finding the hyperparameters of an inverse gamma distribution for σ^2 (and thus a gamma distribution for τ) that was a good visual match to $LN(4.96, 2.70^2)$, obtaining $(\alpha = 0.25, \beta = 0.12)$.

BUGS *results*. At this point I was ready for a first try at results. I chose a burn-in of 1,000 and a short monitoring run of 4,000, obtaining MCMC output that passed the CODA Heidelberger-Welch tests but which had first-order serial correlations of (0.31, 0.58, 0.95) for (μ, σ, ν), leading to default $\hat{n}_{RL}$ values of (4.2K, 6.0K, 67K), respectively. Evidently (a discretized version of) Gibbs sampling is not mixing very well on the degrees of freedom parameter with this data set. So I reran BUGS with the `.cmd` file in Section 7 of Appendix 2, using a burn-in of 1,000 and a monitoring run of 70K (storing every 14th iterate), and obtained results that both yielded good $\hat{n}_{RL}$ values and agreed up to Monte Carlo noise with those in Table 2.11 and Figure 2.13. The only problem is that this second BUGS run took *95* minutes at 333Mhz, versus 7 minutes for the generic Metropolis approach to achieve the same MCMC accuracy: discretization really slows BUGS down. You can show (Problem 2.14) that a from-scratch Gibbs sampler in this problem is considerably more competitive with Metropolis; on the other hand, writing a `.bug` file is considerably easier than programming up your own sampler from the beginning.

If you are fairly new to MCMC, I encourage you both (a) to give BUGS a chance in a number of other problems—as their examples documentation shows, when you stick with conjugate-style priors the BUGS success stories include problems in random-effects logistic and Weibull regression, extra-Poisson variation, latent class models, predictor-variable measurement error, order constraints, changepoints, spatial smoothing, and genetic pedigree analysis—and (b) to write a number of your own samplers from scratch, to develop your intuition about which MCMC strategy is most likely to get you to the finish line most quickly in the applications of principal interest to you.

2.7 Additional reading

[xx this section is incomplete] Gamerman (1997) and lots of references therein; various chapters in Gilks et al. (1996); Gelman et al. chapter 11; Carlin and Louis chapter 8; the MCMC preprint library; **manuscript readers**: please let me know of any important MCMC references I have omitted (bearing in mind the nature of the material presented here).

2.8 Problems

[xx this section is still quite rough]

2.1 [xx sensitivity analysis on effects of outliers in ammonite data]

2.2 ($\mathcal{N}1$) [xx conjugate analysis of the uniform model is possible if one of the two parameters is known but not if both are unknown]

2.3 [xx if $(\theta_t, t = n_B + 1, \ldots, n_B + n_M)$ is a valid sample from the posterior for θ, then $[f(\theta_t), t = n_B + 1, \ldots, n_B + n_M]$ is a valid sample from the posterior for $f(\theta)$ for all reasonable f]

2.4 [xx try Hastings and/or Metropolis out on the ammonite problem]

2.5 [xx Explain how Gibbs fits in with Hastings and Metropolis in the overall MCMC picture]

2.6 [xx figure out induced prior on (A, B) in the ammonite problem – reasonable?]

2.7 [xx sensitivity analysis on specification of hyperparameters L and H in the ammonite problem]

2.8 [xx Standard situation in which the full conditionals are recognizable and easy to sample from]

2.9 [xx Another standard situation in which the full conditionals are recognizable and easy to sample from]

2.10 [xx show that in model (2.30, 2.31) the MLEs are as advertised]

2.11 [xx try strategy (1–2′–4′) on the NB10 data]

2.12 [xx sensitivity analysis of the prior specification in NB10 case study]

2.13 [xx Explain why the apparently optimal value of ν in the sense of Figure 2.14 is not particularly well supported by the posterior for ν summarized in Table 2.11]

2.14 [xx Show that a from-scratch Gibbs sampler in the NB10 problem is considerably more competitive with Metropolis]

2.15 [xx Express the t model hierarchically as a scale mixture of normals, draw the DAG, and explain the conditional independence relationships]

2.16 [xx Standard situation in which both {Metropolis or Hastings} from scratch and `BUGS` are reasonably straightforward. Monitor yourself in human and computer time to see how long it takes you to get (what should be) similar answers. Also contrast the amount of incremental learning arising from both strategies.]

2.17 ($\mathcal{N}2$) [xx Try simulated annealing (Note 9) on the NB10 problem as an alternative way to find the posterior mode.]

2.9 Notes

2.1 I am grateful to Rob Weiss for drawing my attention to this article, and to Dimitris Fouskakis for digitizing Figure 1 from it.

2.2 A **Poisson process** with intensity λ (e.g., Feller, 1968 and/or Ross, 1970) is a stationary, continuous-time, positive-integer-valued stochastic process $N(t)$ which (conceptually) counts the number of occurrences of something of interest to you in the time interval $[0, t]$ (so that $N(0) = 0$), and which satisfies the following:

- $\{N(t), t \geq 0\}$ has *independent increments*, meaning that for all $t_0 < t_1 < \cdots < t_n$ the quantities $\{[N(t_i) - N(t_{i-1})], i = 1, \ldots, n\}$ are independent for all $n \geq 1$; and
- For all s and t, the number $[N(t+s) - N(s)]$ of occurrences in any interval of length t has a Poisson distribution with mean λt.

Several strong conclusions about $N(t)$ immediately arise from these strong assumptions—for instance, the interarrival times are exponential, and given that $N(t) = n$, the n arrival times have the same distribution as the order statistics of a sample of size n from the $U(0, t)$ distribution.

2.3 Two other approaches worth mentioning are

- **Reference analysis** (Bernardo, 1979), which tries to develop highly diffuse priors and straightforward updating strategies for as wide a variety of standard likelihoods as possible. However (Bernardo and Smith, 1994), this approach has trouble with multiparameter problems, hierarchical models, and prediction, rendering it less general than the other methods on which I focus; and
- **Numerical quadrature** (Smith et al., 1985), which uses ideas from the standard numerical analysis literature on quadrature (Bayesians are not the only people who have to evaluate high-dimensional integrals, after all) modified to the Bayesian context. People working in this area report considerable success in models with small k (less than about 7),

but the approach seems problematic with a large number of parameters.

2.4 The basic idea (e.g., Bernardo and Smith, 1994) relies on simple Taylor series calculations. In the continuous case, for instance, with θ a k-vector of parameters and y an n-vector of univariate outcomes, write $p(\theta|y)$ as proportional to $\exp\{\log[p(\theta)] + \log[l(\theta|y)]\}$; expand each of the log terms inside the $\exp\{\cdot\}$ about their respective maxima, keeping only the constant, linear (which vanish), and quadratic bits; and collect like terms together. Then under relatively mild regularity conditions guaranteeing that the remainder terms go to 0 with increasing n (e.g., LeCam and Yang, 1990), $p(\theta|y)$ should be close for large n to a multivariate normal distribution with mean vector

$$H_n^{-1}\left[H_0\,\theta_0 + H(\hat{\theta})\,\hat{\theta}\right] \tag{2.50}$$

and k by k covariance matrix

$$H_n^{-1} = \left[H_0 + H(\hat{\theta})\right]^{-1}, \tag{2.51}$$

where H_0 is -1 times the Hessian (matrix of second derivatives) of the log prior evaluated at the prior mode θ_0 and $H(\hat{\theta})$ is -1 times the Hessian of the log likelihood evaluated at the MLE $\hat{\theta}$:

$$H_0 = \left(-\frac{\partial^2 \log[p(\theta)]}{\partial\theta_i \partial\theta_j}\right)_{\theta=\theta_0}, \quad H(\hat{\theta}) = \left(-\frac{\partial^2 \log[l(\theta|y)]}{\partial\theta_i \partial\theta_j}\right)_{\theta=\hat{\theta}}. \tag{2.52}$$

Notice how similar these expressions are to the corresponding formulae (1.26) and (1.28), in the simple model from Chapter 1 with Gaussian prior and likelihood for an unknown μ: the posterior mean is a weighted average of a prior measure of center θ_0 and a data measure of center $\hat{\theta}$, weighted by the multivariate analogue of their respective precisions, and the posterior precision (matrix) H_n is the sum of the prior (H_0) and data ($H(\hat{\theta})$) precisions.

2.5 The main exception I am thinking of is the beautiful Laplace-style investigation conducted by Mosteller and Wallace (1964, 1984) into the authorship of the *Federalist* papers.

2.6 Intuitively speaking, a (discrete-time) **Markov chain** (e.g., Feller, 1968; Roberts, 1996; Gamerman, 1997) is a stochastic process unfolding in time in such a way that the past and future

states of the process are independent given the present state—in other words, to figure out where the chain is likely to go next you don't need to pay attention to where it's been, you just need to consider where it is now. More formally, a stochastic process $\{\theta_t, t \in T\}$, $T = \{0, 1, \ldots\}$, with state space S is *Markov* if, for any set $A \in S$,

$$P(\theta_{t+1} \in A|\theta_0, \ldots, \theta_t) = P(\theta_{t+1} \in A|\theta_t). \tag{2.53}$$

The theory of Markov chains is harder mathematically if S is continuous (e.g., Tierney, 1996), which is what we need for MCMC with real-valued parameters, but most of the main ideas emerge with discrete state spaces, and I will assume discrete S in the intuitive discussion below. Generalizations to continuous time are also possible (e.g., Feller, 1971) but are not relevant here.

The idea in MCMC is (a) to set things up so that the Markov chain converges to an *equilibrium* or *stationary* distribution, and (b) to further contrive that this distribution is $p(\theta|y)$. To achieve the first goal, the chain needs to satisfy three properties:

- It must be *irreducible*, which basically means that no matter where it starts the chain has to be able to reach any other state in a finite number of iterations with positive probability;
- It must be *aperiodic*, meaning that for all states i the set of possible *sojourn times*, to get back to i having just left it, can have no divisor bigger than 1. This forces the chain to mix freely among its possible states rather than oscillating back and forth within a subset of S; and
- It must be *positive recurrent*, meaning that (a) for all states i, if the process starts at i it will return to i with probability 1, and (b) the expected length of waiting time til the first return to i is finite. Notice that this is a bit delicate: wherever the chain is now, we insist that it must certainly come back here, but we don't expect to have to wait forever for this to happen.

A positive recurrent and aperiodic chain is called *ergodic*, and it turns out that such chains possess a unique *stationary* (or *equilibrium*, or *invariant*) distribution π, characterized by the relation

$$\pi(j) = \sum_i \pi(i) P_{ij}(t) \tag{2.54}$$

for all states j and times $t \geq 0$, where $P_{ij}(t) = P(\theta_t = j | \theta_0 = i)$ is the *transition matrix* of the chain. Informally, the stationary distribution characterizes the behavior that the chain will settle into after it has been run for a long time, regardless of its initial state.

The MCMC point of having set up all this machinery is the *ergodic theorem*: if $\{\theta_t\}$ is ergodic and f is any real-valued function for which $E_\pi |f(\theta)|$ is finite, then with probability 1

$$\frac{1}{n_M} \sum_{t=1}^{n_M} f(\theta_t) \to E_\pi [f(\theta)] = \sum_i f(i)\pi(i), \tag{2.55}$$

in which the right side is just the expectation of $f(\theta)$ under the stationary distribution π. In plain English this means that—as long as the stationary distribution *is* $p(\theta|y)$ (see the next endnote)—you can learn (to arbitrary accuracy) about things like posterior means, SDs, and so on just by waiting for stationarity to kick in and monitoring thereafter for a long enough period. Of course, as Roberts (1996) notes, the theorem is silent on the two key practical questions it raises: how long you have to wait for stationarity, and how long to monitor after that (Sections 2.3 and 2.4).

2.7 We may as well look at the stationary distribution in the case of a continuous state space S, for instance $\mathbb{R}^k$. As noted (for example) by Tierney (1996), to pin down the distribution of a Markov chain $\theta^{(t)}$ when S is continuous, you need to know two things: its initial distribution across the states in S, and its *transition kernel*, the continuous analogue of the transition matrix in Note 6: for any $\theta \in S$, $A \subset S$, and $t \geq 0$, this is the function $P(\theta, A) = P(\theta_{t+1} \in A | \theta_t = \theta)$. The transition kernel just specifies the distribution of the chain's location at time $(t+1)$ given that it was at θ at time t.

The argument (one version of it, at least; e.g., Gilks et al., 1996b) for deriving the stationary distribution proceeds in four steps. (1) By looking at the Hastings algorithm in (2.6) and thinking about the possible moves at any given time t, you can see that the transition kernel of the Hastings sampler satisfies the equation

$$P(\theta_{t+1}|\theta_t) \quad = \quad f(\theta_{t+1}|\theta_t)\, \alpha_H(\theta_t, \theta_{t+1}) + I(\theta_{t+1} = \theta_t) \cdot$$

$$\left[1 - \int f(\theta|\theta_t)\, \alpha_H(\theta_t, \theta)\, d\theta\right], \qquad (2.56)$$

where $I(\cdot)$ is the indicator function (the first right-side term in (2.56) picks up the possibility that the chain moves, and the second that it stays put). (2) If you expand out the definitions of both $\alpha_H(\theta_t, \theta_{t+1})$ and $\alpha_H(\theta_{t+1}, \theta_t)$ (by which I mean, for instance, $\alpha_H(\theta_t, \theta_{t+1}) =$ something if such-and-such is true and something else if not) and form the ratio $\frac{\alpha_H(\theta_t, \theta_{t+1})}{\alpha_H(\theta_{t+1}, \theta_t)}$, you will see that (Metropolis et al. and) Hastings picked the acceptance probabilities so that

$$\frac{\alpha_H(\theta_t, \theta_{t+1})}{\alpha_H(\theta_{t+1}, \theta_t)} = \frac{p(\theta_{t+1}|y) f(\theta_t|\theta_{t+1})}{p(\theta_t|y) f(\theta_{t+1}|\theta_t)}. \qquad (2.57)$$

(3) This, together with (2.56) and some algebra, shows that the chain satisfies the *detailed balance equation*,

$$p(\theta_t|y)\, P(\theta_{t+1}|\theta_t) = p(\theta_{t+1}|y)\, P(\theta_t|\theta_{t+1}), \qquad (2.58)$$

which is the crucial thing that gives what we want: (4) Integrating (2.56) over the possible values of θ_t and plugging in detailed balance yields

$$p(\theta_{t+1}|y) = \int p(\theta_t|y)\, P(\theta_{t+1}|\theta_t)\, d\theta_t, \qquad (2.59)$$

which is the continuous-state-space version of (2.54). This has demonstrated that, *if* the Markov chain created by the Hastings algorithm (2.6) has a stationary distribution, then that distribution must be $p(\theta|y)$; see Tierney (1996) for details and precise conditions that ensure convergence. **NB** Detailed balance is closely related to *reversibility* of the chain: in the language used here, a Markov chain is reversible if it is positive recurrent with stationary distribution $p(\cdot|y)$ satisfying the detailed balance condition (2.58).

2.8 This terminology is slightly nonstandard. Most people talk about Metropolis-Hastings sampling without specifying in the name whether the proposal distribution is symmetric (Metropolis; see (2.21)) or not (Hastings), but I will often retain the distinction in what follows.

2.9 **Simulated annealing** (SA; e.g., Geman and Geman 1984) is a stochastic optimization method for maximizing a (nearly) arbitrary real-valued function $f(\theta)$ ($\theta \in \mathbb{R}^k$), based on a nice idea

that fits in well with the other MCMC approaches in this chapter. If f is unimodal then any standard method should find the mode without much trouble, for instance Newton-Raphson from even a not-very-good starting point, so to make things tougher suppose f has one or more local maxima in addition to the global one.

Algorithm (*simulated annealing*). To maximize a posterior distribution $p(\theta|y)$, choose a proposal distribution (PD) $f(\theta|\theta_t)$ and a *cooling schedule* T_t, define

$$\alpha_{SA}(\theta_t, \theta^*) = \exp\left\{-\frac{\log[p(\theta_t|y)] - \log[p(\theta^*|y)]}{T_t}\right\},$$

and

```
Initialize θ0 ; t ← 0
Repeat {
  Sample θ* ~ f(θ|θt)
  If p(θ*|y) > p(θt|y) then θt+1 ← θ*
    else {
      Sample u ~ U(0,1)
      If u ≤ αSA(θt, θ*) then θt+1 ← θ*
        else θt+1 ← θt
    }
  t → (t + 1)
}
```

(2.60)

A greedy stochastic hill-climbing strategy in this situation might proceed like this: at time t in the search, (a) generate a new candidate place to consider moving to, say θ^*, and (b) compare $f(\theta_t)$ and $f(\theta^*)$. If the new place is better (higher) then move there (set $\theta_{t+1} = \theta^*$); otherwise discard this θ^*, go back to (a), generate another candidate, and so on. With high likelihood this will eventually get you to the top of the nearest hill, but once you are there it won't allow you to jump away from this hill and find a higher peak (if any) somewhere else. SA improves on this by allowing you to sometimes go downhill (early on in the search process, at least), in the hope that by temporarily making things worse you will eventually wander to the highest place of all. SA implements this by using a rule of the form {if θ^* is better, then by all means move there, but if it's worse, move there anyway

with probability α_{SA}}. Formally, the algorithm, in the context of maximization of a posterior $p(\theta|y)$, is summarized in (2.60).

The idea behind the *cooling schedule* in SA is the following. Imagine you were wandering around in the plane (this is like generating proposed places to move when θ has $k = 2$ components) looking for the maximum of $p(\theta|y)$, which is (let's say) highly concentrated in a small region, and you were far from that region—the higher the peak was, the easier it would be for you to spot it. This suggests trying to maximize an exaggerated or heightened version of the posterior, for instance $[p(\theta|y)]^{\frac{1}{T}}$ for $T < 1$, instead of p itself, and the closer T is to 0 the more exaggerated $[p(\theta|y)]^{\frac{1}{T}}$ will be. So while you are letting the iteration counter t run from 1 to n_M (say), making new proposed moves θ^* all the while, why not let T get smaller and smaller as a function of t as well? T_t is called the *temperature* parameter in the SA algorithm, and any method for monotonically decreasing it from a starting value T_0 (1.0, say) to a final value T_f (0.001, say) is called a cooling schedule. There are a number of possibilities; one that often seems to work well (Stander and Silverman, 1994) is a geometric decline, $T_t = T_0\gamma^t$ for $\gamma = \left(\frac{T_f}{T_0}\right)^{\frac{1}{n_M}}$. To find the global mode you run the algorithm repeatedly, each time with a large n_M (like 10,000), and using a (widely dispersed) variety of starting values θ_0, possibly also varying T_0. Early on in these runs α_{SA} will be fairly large and you will often jump to locally inferior places, but as T_t approaches 0 so does α_{SA} and the process eventually "freezes" at one particular mode. There is no guarantee that this is the global max (which is why you should run it with a number of different θ_0), but SA's willingness to go downhill as well as up often allows it to out-perform greedier search methods at "bump-hunting."

As I mentioned in the main text, I like SA better than the Gelman-Rubin strategy for finding multiple modes, for the following reason. A bit of algebra should convince you that if, instead of varying the temperature, you hold T constant at 1, the acceptance probability α_{SA} *coincides with* α_M*, the acceptance rate (2.21) from Metropolis sampling: indeed when* $T = 1$*, SA and Metropolis are identical*. Thus you can solve two important problems with one piece of software by writing an SA routine—cool the process by sending $T \downarrow 0$ to find the mode(s), or hold

the temperature constant at 1 to extract the usual posterior marginal and predictive summaries with Metropolis.

2.10 The (sample) **autocorrelation** function (ACF) for a time series θ_t (e.g., Box and Jenkins, 1976; Chatfield, 1996) simply measures the degree to which knowledge of the past of the series is linearly predictive of its future. Specifically, this function is given by

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k}(\theta_t - \bar{\theta})(\theta_{t-k} - \bar{\theta})}{\sum_{t=1}^{n}(\theta_t - \bar{\theta})^2} \tag{2.61}$$

as k, measuring the number of *lags* backwards in the series, varies from $0, \ldots, n-1$, where n is the number of observed time points and $\bar{\theta} = \frac{1}{n}\sum_{t=1}^{n}\theta_t$. The *first-order* autocorrelation (or *serial* correlation) $\hat{\rho}_1$ is often the star of the show in MCMC work, because the columns in the MCMC data set often behave a lot like *autoregressive* processes of order 1 (see the next endnote).

2.11 The **autoregressive process** of order 1 with lag-1 serial correlation ρ (e.g., Box and Jenkins, 1976, Chatfield, 1996), abbreviated $AR_1(\rho)$, is modeled as follows:

$$\theta_t - \mu = \alpha_1(\theta_{t-1} - \mu) + z_t, \tag{2.62}$$

where the z_t are *white noise*, assumed IID $N(0, \sigma_z^2)$. Decent approximate estimates of the mean and regression parameters are given by the intuitively obvious $\hat{\mu} = \bar{\theta}$ and $\hat{\alpha}_1 = \hat{\rho}_1$. The autocorrelation function of an AR_1 process is $\rho_k = \rho_1^k$—in other words, for positive ρ_1 a plot of the sample autocorrelations should show a steady geometric decay—and the partial autocorrelation function (PACF; endnote 20) has a spike of height ρ_1 at lag 1 and is zero thereafter—so the sample ACF and PACF should (in theory) make diagnosing an AR_1 process pretty straightforward.

2.12 This makes the sub-chain, observed only at the times at which a move actually takes place, a *martingale* (e.g., Breiman, 1968), a fact that helps to establish some useful properties of MCMC samplers.

2.13 For many purposes in working with the SI-χ^2 distribution, the factor $(\sigma_*^2)^{\frac{\nu^*}{2}}$ in (2.14) can be treated as a throw-away constant, but not when (2.15) or something like it is used as a PD: to compute the acceptance probability (2.5) the log PD has to be evaluated with arguments (σ^2, σ_t^2) half the time and (σ_t^2, σ^2) the

other half, so that the term $(\sigma_*^2)^{\frac{\nu^*}{2}}$ does not cancel in evaluating (2.5). The first time I programmed up the Hastings sampler for this model I made the mistake of ignoring this factor and got results in which the posterior mean depended on ν^*, which of course cannot happen if the implementation is correct. I am indebted to Bill Browne for helping me spot this error.

2.14 All timings in this book were made on a dedicated `DECalpha` Unix workstation running at 333Mhz, or on one or another of a variety of Unix `SPARCstation` and `UltraSPARC` CPU servers (with the appropriate conversion in timings made to 333Mhz).

2.15 You can show (e.g., Bernardo and Smith, 1994) that the predictive distribution for y^* given y in this model is a scaled t, with degrees of freedom $(\nu_p + n)$, mean μ, and scale parameter $\frac{\nu_p \sigma_p^2 + n s_*^2}{\nu_p + n}$.

2.16 Another option is to write the slow bits of your program in `Fortran` or `C` and call them from within `S+`; see Venables and Ripley (1997).

2.17 There is a direct analogy between the form and scale of a proposal distribution in MCMC and the choice of kernel and window width in density estimation, and the same results (e.g., Silverman, 1986) apply: it doesn't matter too much whether you use (say) a Gaussian or uniform PD (kernel); what matters a lot is to get the PD scale (window width) right.

2.18 I actually computed the density estimate on the $\log(\sigma^2)$ scale and back-transformed it, and I used a big window width, namely $0.25[\max\{\log(\sigma^2)\} - \min\{\log(\sigma^2)\}]$. The `S+` code is

```
d <- density( log( sigma2 ), width = ( max( log(
  sigma2 ) ) - min( log( sigma2 ) ) ) / 4 )
plot( exp( d$x ), d$y, type = 'l', xlab = 'sigma2',
  ylab = 'Density', xlim = c( 25, 75 ) )
```

2.19 The **partial autocorrelation function** (PACF) ϕ_{kk} at lag k of a time series measures "the excess correlation not accounted for by an AR_{k-1} model" (Chatfield, 1996), and may be estimated by "successively fitting AR_p processes for $p = 1, 2, \ldots$ and picking out the estimates of the last coefficients fitted at each stage" (Box and Jenkins, 1976). Thus the PACF is a kind of direct diagnostic for AR_p processes: if something is (say) AR_1 then there will be no "excess correlation not accounted for by an AR_1

model," so that the PACF at lag 2 and thereafter will be zero. Specifically, for AR_p models, $\phi_{11} = \rho_1$ and $\phi_{22} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}$, so that $\phi_{22} = 0$ for an AR_1 process (because for such a process $\rho_2 = \rho_1^2$; see endnote 11). When least squares or something equivalent is used to estimate the ϕ_{kk}, you can show that $SE(\hat{\phi}_{kk}) = \frac{1}{\sqrt{n}}$ for $k \geq p + 1$ when the series is AR_p (and a similar result applies for the ACF), which explains the little horizontal dotted lines in Figure 2.6: it is natural, in diagnosing the order of an AR_p process from the PACF, to look for the first lag $k = p + 1$ for which $\hat{\phi}_{kk}$ and all higher partial autocorrelations do not differ significantly from 0, and conclude tentatively that the series is AR_p. On this basis both the ACF and PACF in Figure 2.6 are nearly perfect AR_1.

2.20 [xx details on Heidelberger-Welch]

2.21 Some of the results in this subsection are joint with Bill Browne, who has more extensive findings of this type in his PhD dissertation, Browne (1998).

2.22 Of course, as usual in the design of sampling experiments, if I decide that I can't afford 248K iterations I can take solace in less stringent requirements. For instance, if all I want is to make sure that the Monte Carlo standard error for the posterior mean of σ^2 is 0.05, then I get to divide 248K by 1.96^2, producing the more pleasant-sounding target of 65K iterations, and if I relax d further from 0.05 to 0.1 I can divide again by $\left(\frac{0.1}{0.05}\right)^2$ to obtain a requirement of 16K iterations, which brings me into the Raftery-Lewis default neighborhood. Also, to be fair, this is Hastings, not Gibbs (see Section 2.5), where the serial correlations are usually lower.

2.23 This can be shown directly, using reasoning like that laid out in Note 7, but the similarity between Gibbs and Metropolis-Hastings (MH) should suggest an easier route—if you could show that Gibbs is a special case of MH, then you would be done. But this is not hard, as follows. As I will examine in Section 2.6, given a parameter vector θ of dimension k, you could either make an MH update on all of θ at once, or you could divide θ into subcomponents or **blocks** (which may have only one element of θ in them) and update the blocks sequentially. The latter approach, which people call **single-component** Metropolis-Hastings, was in fact the one originally proposed by Metropolis et al. Once a

series of proposal distributions for each block is specified, possibly conditional on the current results in some or all of the other blocks, it is straightforward (see, e.g., Gilks et al., 1996b) to show that Gibbs is just a special case of single-component MH in which the acceptance probability is always 1.

2.24 This is the same idea on which the Shapiro-Wilk (1965) test for normality is based: essentially they compute the squared correlation between the ordered data values and the expected quantiles of a standard normal, which is just a numerical summary of the fit to the Gaussian as diagnosed by the usual normal qqplot.

2.25 If the log density of the full conditional is sufficiently smooth to have a second derivative, log concavity means that this derivative should be everywhere non-negative. Gilks (1992), who developed adaptive-rejection sampling with Wild (Gilks and Wild, 1992), actually requires less than this: only that if you pick three values θ_i in the support of the full conditional $p(\theta)$ (say), with $\theta_1 < \theta_2 < \theta_3$, and define the points $P_i = [\theta_i, p(\theta_i)]$, the gradient of the chord joining P_2 and P_3 can be no larger than that of the chord joining P_1 and P_2.

2.26 In fact log-concavity holds in this model for both the full conditionals for σ and ν, but `BUGS` seems unable to verify this fact.

CHAPTER 3

Hierarchical models for combining information

CHAPTER 4

Hierarchical model diagnostics

CHAPTER 5

Random-effects and mixed models

CHAPTER 6

Longitudinal data analysis

CHAPTER 7

Mixture modeling

CHAPTER 8

Hierarchical modeling as an approach to model selection

8.1 Model expansion

work in mixtures of conjugate priors (Diaconis and Ylvisaker 19xx);

8.2 Bayes factors and Laplace approximations

8.3 The effects of model uncertainty

8.4 Case study: Effects of an intervention to reduce hospitalization rates for elderly people

8.5 Case study: Risk assessment in the *Challenger* space shuttle disaster

8.6 Additional reading

8.7 Problems

1.1 simple example $\alpha \sim U(\alpha_{\text{lo}}, \alpha_{\text{hi}}$ in ch. 1 mortality example, etc.

8.8 Notes

CHAPTER 9

Discussion and further topics

Appendix 1: Some common prior and likelihood families

Continuous: beta, normal, $t_\nu(\mu, \sigma^2)$, $SI\text{-}\chi^2(\nu_0, \sigma_0^2)$ (Gelman et al., 1995)

Discrete: Bernoulli, binomial

Appendix 2: Software details

1 A Hastings sampler in S+ for the Gaussian example in Section 2.2

If you want to try this yourself (highly recommended if you are a newcomer to MCMC), the functions below are available (a) on the web at `http://www.bath.ac.uk/~masdd` or (b) by anonymous `ftp`, as follows: `ftp` to `ftp.bath.ac.uk`, type `anonymous` when it asks for your name, type your email address when it asks for your password, and issue the commands `cd pub/masdd/Papers` and `get bhm-code.t`.

```
# S+ functions to do Hastings sampling in the model (2.7):
#
#   sigma2 ~ SI-chisq( nu.p, sigma2.p )
#   ( y_i | sigma2 ) ~IID N( mu, sigma2 ), i = 1, ..., n
#
# (written by DD, with trick to avoid S+ memory-
#   management problems due to Brian Ripley)
#
# Inputs:
#
#   y = data vector, of length n = sample size
#   mu = known mean in Gaussian likelihood
#   nu.p = prior effective sample size
#   sigma2.p = prior estimate of sigma2
#   sigma2.0 = initial value for sigma2 in Hastings iterations
#   nu.star = scaling factor for Hastings proposal distribution
#     (affects the acceptance rate R; to increase R, increase
#     nu.star). nu.star must be > 2 in this implementation;
#     values near 20-25 lead to good mixing
#   n.burnin = length of burn-in period
#   n.monitor = length of monitoring period
#   n.thin = thinning constant (only every n.thin-th iteration
#     in the monitoring period will be written to disk)
#   seed = random number seed (for generating repeatable
#     sequences of Hastings iterations); must be an integer
#     from 0 to 1000
#   output.file.prefix = character string naming where you want
#     the MCMC data set to go; for example, output.file.prefix
#     = "NB10" would write the MCMC data set to the file
#     "NB10.d"
```

```
  #
  # Outputs:
  #
  #   Acceptance rate R returned when iterations are finished
  #   A file called paste( output.file.prefix, ".d", sep = "" ) is
  #     written (in the same directory where S+ has been called)
  #     containing one row for each monitored iteration and three
  #     columns: the monitored iteration number (from 1 to
  #     n.monitor / n.thin), the simulated draw from the posterior
  #     for sigma2 for that iteration, and the corresponding
  #     simulated draw from the predictive distribution for a new
  #     y.star. If the output file exists before the function is
  #     invoked, it will be over-written

hastings.gaussian.variance <- function( y, mu, nu.p, sigma2.p,
  sigma2.0, nu.star, n.burnin, n.monitor, n.thin, seed,
  output.file.prefix ) {

  # Main routine
  #
  # Includes trick due to Ripley to overcome S+ memory-management
  #   problems

  sigma2.old <- sigma2.0
  R <- 0
  write( c( sigma2.old, R ), "loop.result", append = F )

  set.seed( seed )

  for ( i in 1:( n.burnin + n.monitor ) ) {

    null <- loop( y, mu, nu.p, sigma2.p, nu.star,
      output.file.prefix, i )

  }

  loop.result <- scan( "loop.result" )
  R <- loop.result[2]
  return( R / n.monitor )

}

loop <- function( y, mu, nu.p, sigma2.p, nu.star,
  output.file.prefix, i ) {

  # Ripley idea: put everything inside an explicit loop into a
  #   function that returns nothing, reading from and writing to
  #   disk as needed to maintain communication. This will keep S+
  #   from accumulating dynamic memory as it goes around the loop.

  loop.result <- scan( "loop.result" )
  sigma2.old <- loop.result[1]
  R <- loop.result[2]
```

```
  sigma2.star <- PD.sim( nu.star, ( nu.star - 2 ) * sigma2.old /
    nu.star )
  u <- runif( 1 )
  b <- ( u <= alpha( sigma2.old, sigma2.star, y, mu, nu.p,
    sigma2.p, nu.star ) )
  sigma2.new <- sigma2.star * b + sigma2.old * ( 1 - b )
  y.new <- rnorm( 1, mu, sqrt( sigma2.new ) )

  if ( i > n.burnin ) R <- R + b
  if ( ( i > n.burnin ) & ( ( i - n.burnin ) %% n.thin == 0 ) )
    write( c( ( i - n.burnin ) / n.thin, signif( c( sigma2.new,
      y.new ), digits = 5 ) ), paste( output.file.prefix, ".d",
      sep = "" ), ncol = 3, append = ( i > n.burnin + n.thin ) )

  sigma2.old <- sigma2.new
  write( c( sigma2.old, R ), "loop.result", append = F )
  return( NULL )

}

PD.sim <- function( nu, sigma2 ) {

  # Proposal distribution simulation

  return( nu * sigma2 / rchisq( 1, nu ) )

}

alpha <- function( sigma2.old, sigma2.new, y, mu, nu.p, sigma2.p,
  nu.star ) {

  # Acceptance probability calculation

  return( min( 1, exp( log.post( sigma2.new, y, mu, nu.p,
    sigma2.p ) + log.PD( sigma2.old, nu.star, sigma2.new ) -
    log.PD( sigma2.new, nu.star, sigma2.old ) - log.post(
    sigma2.old, y, mu, nu.p, sigma2.p ) ) ) )

}

log.post <- function( sigma2, y, mu, nu.p, sigma2.p ) {

  # log( posterior ) calculation

  return( log.lik( sigma2, y, mu ) + log.prior( sigma2, nu.p,
    sigma2.p ) )

}

log.lik <- function( sigma2, y, mu ) {

  # log( likelihood ) calculation

  n <- length( y )
```

```
  return( ( - n / 2 ) * log( sigma2 ) - sum( ( y - mu )^2 ) /
    ( 2 * sigma2 ) )

}

log.prior <- function( sigma2 , nu.p, sigma2.p ) {

  # log( prior ) calculation

  return( ( -1 - nu.p / 2 ) * log( sigma2 ) - nu.p * sigma2.p /
    ( 2 * sigma2 ) )

}

log.PD <- function( sigma2, nu.star, sigma2.star ) {

  # log( proposal distribution ) calculation

  return( ( nu.star / 2 ) * log( sigma2.star ) - ( 1 + nu.star /
    2 ) * log( sigma2 ) - ( nu.star - 2 ) * sigma2.star / ( 2 *
    sigma2 ) )

}
```

2 An S+ function to prepare MCMC output for diagnostic analysis by CODA

The set of S+ functions CODA (highly recommended), for performing a variety of MCMC diagnostic checks and summaries, is available on the web at http://www.mrc-bsu.cam.ac.uk or by anonymous ftp at ftp.mrc-bsu.cam.ac.uk.

```
  # S+ function to prepare the output of all the S+ and C samplers
  #   supplied here for CODA diagnostic processing
  #
  # (written by DD)
  #
  # Inputs:
  #
  #   n.monitor = length of monitoring period used to produce
  #     input file
  #   n.thin = thinning constant used to produce input file
  #   p = number of variables monitored in input file (this is
  #     one less than the number of columns in that file, since
  #     the first column contains the iteration number)
  #   input.file.prefix = character string naming the file with
  #     the MCMC data set in it; for example, input.file.prefix
  #     = "NB10" assumes that the MCMC data set is in a file
  #     called "NB10.d" in the directory where S+ is running
  #   var.names = character vector of length p supplying (in order)
  #     the names of the variables monitored in MCMC data set
  #   output.file.prefix = character string naming where you want
```

```
  #     the output of preCODA to go; for example,
  #     output.file.prefix = "NB10" will write the MCMC data set
  #     out into a file called "NB10.out" in a format suitable for
  #     reading by CODA, and will also create a file called
  #     "NB10.ind" that CODA uses to figure out the format of
  #     "NB10.out"
  #
  # Outputs:
  #
  #   Files called paste( output.file.prefix, ".out", sep = "" )
  #     and paste( output.file.prefix, ".ind", sep = "" )
  #     are written (in the same directory where S+ has been
  #     called); see under output.file.prefix above for a
  #     description of the two files. If these files exist before
  #     preCODA is invoked, they will be over-written

preCODA <- function( n.monitor, n.thin, p, input.file.prefix,
  var.names, output.file.prefix ) {

  # Prepares MCMC output for reading by CODA

  MCMC.data <- matrix( scan( paste( input.file.prefix, ".d",
    sep = "" ) ), n.monitor / n.thin, p + 1, byrow = T )

  for ( i in 1:p ) {

    write( t( cbind( MCMC.data[, 1], MCMC.data[, i + 1] ) ),
      paste( output.file.prefix, ".out", sep = "" ), ncol = 2,
      append = ( i > 1 ) )
    write( c( var.names[i], as.character( 1 + ( i - 1 ) *
      n.monitor / n.thin ), as.character( i * n.monitor /
      n.thin ) ), paste( output.file.prefix, ".ind", sep = "" ),
      ncol = 3, append = ( i > 1 ) )

  }

  return( paste( output.file.prefix, ".out", " written", sep =
    "" ) )

}
```

3 A Hastings sampler in C for the Gaussian example in Section 2.2

```
/*
 * C functions to do Hastings sampling in the model (2.7):
 *
 *   sigma2 ~ SI-chisq( nu.p, sigma2.p )
 *   ( y_i | sigma2 ) ~IID N( mu, sigma2 ), i = 1, ..., n
 *
 * (random number generators written by William Browne; Hastings
 *   code written by DD and Dimitris Fouskakis and edited by WB)
 *
```

```
 * Inputs are contained in a file called
 *   "hastings.gaussian.variance.in", in the directory where the
 *   program is to be run, with the contents listed below (one
 *   input per line, in the order listed, from mu to n; then the
 *   data vector y is input, with one element on each line)
 *
 *   mu = known mean in Gaussian likelihood
 *   nu.p = prior effective sample size
 *   sigma2.p = prior estimate of sigma2
 *   sigma2.0 = initial value for sigma2 in Hastings iterations
 *   nu.star = scaling factor for Hastings proposal distribution
 *     (affects the acceptance rate R; to increase R, increase
 *     nu.star). nu.star must be > 2 in this implementation;
 *     values near 20-25 lead to good mixing
 *   n.burnin = length of burn-in period
 *   n.monitor = length of monitoring period
 *   n.thin = thinning constant (only every n.thin-th iteration
 *     in the monitoring period will be written to disk)
 *   seed = random number seed (for generating repeatable
 *     sequences of Hastings iterations); must be an integer
 *   n = sample size = length of data vector
 *   y = data vector
 *
 * Outputs:
 *
 *   Acceptance rate R printed when iterations are finished
 *   A file called "hastings.gaussian.variance.d" is written (in
 *     the same directory where the program has been run)
 *     containing one row for each monitored iteration and three
 *     columns: the monitored iteration number (from 1 to
 *     n.monitor / n.thin), the simulated draw from the posterior
 *     for sigma2 for that iteration, and the corresponding
 *     simulated draw from the predictive distribution for a new
 *     y.star. If "hastings.gaussian.variance.d" exists before
 *     the function is invoked, it will be over-written
 */

#include <stdio.h>
#include <math.h>

/*
 * Defined constants
 */
#define SEED1 13
#define SEED2 4
#define SEED3 1972
#define PI 3.1415927
#define E 2.71828182

long int seed1;  /* Seeds declared externally to avoid passing */
long int seed2;  /* each time wichmann is called               */
long int seed3;

double wichmann()
```

```
/*
 * Random number generator for U(0,1) distribution.
 */
{
   extern long int seed1, seed2, seed3;
   double random;
   seed1 = (171 * seed1)%30269;
   seed2 = (172 * seed2)%30307;
   seed3 = (170 * seed3)%30323;

   random = fmod(seed1/30269.0 + seed2/30307.0 + seed3/30323.0,
     1.0);
   return random;
}

double rexp(double lambda)
/*
 * Generates from an exponential distribution
 */
{
   double random, uniform;
   uniform = wichmann();
   random = - (1/lambda) * log(uniform);
   return random;
}

double rgamma1(double alpha)
/*
 * Generates from a gamma distribution with alpha < 1
 */
{
   double uniform0, uniform1;
   double random, x;
   int done = 0;
   uniform0 = wichmann();
   uniform1 = wichmann();
   if (uniform0 > E/(alpha + E))
   {
      random = -log((alpha + E)*(1-uniform0)/(alpha*E));
      if ( uniform1 > pow(random,alpha - 1))
         return -1;
      else
         return random;
   }
   else
   {
      x = (alpha + E) * uniform0 / E;
      random = pow(x,1/alpha);
      if ( uniform1 > exp(-random))
         return -1;
      else
         return random;
   }
}
```

```
double rgamma2(double alpha)
/*
 * Generates from a gamma distribution with alpha > 1
 */
{
   double uniform1,uniform2;
   double c1,c2,c3,c4,c5,w;
   double random;
   int done = 1;
   c1 = alpha - 1;
   c2 = (alpha - 1/(6 * alpha))/c1;
   c3 = 2 / c1;
   c4 = c3 + 2;
   c5 = 1 / sqrt(alpha);
   do
   {
      uniform1 = wichmann();
      uniform2 = wichmann();
      if (alpha > 2.5)
      {
          uniform1 = uniform2 + c5 * (1 - 1.86 * uniform1);
      }
   }
   while ((uniform1 >= 1) || (uniform1 <= 0));
   w = c2 * uniform2 / uniform1;
   if ((c3 * uniform1 + w + 1/w) > c4)
   {
      if ((c3 * log(uniform1) - log(w) + w) >= 1)
      {
         done = 0;
      }
   }
   if (done == 0)
      return -1;
   random = c1 * w;
   return random;
}

double rgamma(double alpha, double beta)
/*
 * Generates from a general gamma(alpha,beta) distribution
 */
{
   double random;
   if (alpha < 1)
      do {
      random = rgamma1(alpha)/beta;
      } while (random < 0 );
   if (alpha == 1)
      random = rexp(1)/beta;
   if (alpha > 1)
      do {
      random = rgamma2(alpha)/beta;
```

```
      } while (random < 0);
   return random;
}

double rstd_normal()
/*
 * Generates from a standard normal(0,1) distribution
 */
{
   double uniform1,uniform2;
   double theta,r;
   double random;
   uniform1 = wichmann();
   uniform2 = wichmann();
   theta = 2 * PI * uniform1;
   r = sqrt(2 * ( - log(uniform2)));
   random = r * cos(theta);
   return random;
}

double rnormal(double mean, double sd)
/*
 * Generates from a general normal(mu,sigma2) distribution
 */
{
   double random;
   random = mean + sd * rstd_normal();
   return random;
}

/*
 * Hastings code begins here
 */

double PD_sim( nu, sigma2 )
  double nu, sigma2;
/*
 *  Proposal distribution simulation
 */
{
  double result = nu * sigma2 / rgamma( nu / 2.0, 0.5 );
  return result;
}

double log_lik( sigma2, y, mu, n)
  double sigma2, *y, mu;
  long int n;
/*
 *  log( likelihood ) calculation
 */
{
  int i;
  double result = ( - n / 2.0 ) * log( sigma2 );
  for ( i = 0; i < n; i++ )
```

```
  {
    result = result - pow( y[i] - mu, 2.0 ) / ( 2.0 * sigma2 );
  }
  return result;
}

double log_prior( sigma2 , nu_p, sigma2_p )
  double sigma2 , nu_p, sigma2_p;
/*
 *  log( prior ) calculation
 */
{
  double result = ( -1.0 - nu_p / 2.0 ) * log (sigma2 ) -
    nu_p * sigma2_p / ( 2.0 * sigma2 );
  return result;
}

double log_post( sigma2, y, mu, nu_p, sigma2_p, n )
  double sigma2, *y, mu, nu_p, sigma2_p;
  long int n;
/*
 *  log( posterior ) calculation
 */
{
  double result = log_lik( sigma2, y, mu, n ) +
    log_prior( sigma2, nu_p, sigma2_p );
  return result;
}

double log_PD( sigma2, nu_star, sigma2_star )
  double sigma2, nu_star, sigma2_star;
/*
 *  log( proposal distribution ) calculation
 */
{
  double result = ( nu_star / 2.0 ) * log( sigma2_star ) -
    ( 1.0 + nu_star / 2.0 ) * log( sigma2 ) -
    (nu_star - 2.0) * sigma2_star / ( 2.0 * sigma2 );
  return result;
}

double alpha( sigma2_old, sigma2_new, y, mu, nu_p, sigma2_p,
  nu_star, n )
  double sigma2_old, sigma2_new, *y, mu, nu_p, sigma2_p, nu_star;
  long int n;
/*
 *  acceptance probability calculation
 */
{
  double ratio = exp( log_post( sigma2_new, y, mu, nu_p, sigma2_p,
    n ) + log_PD( sigma2_old, nu_star, sigma2_new ) -
    log_PD( sigma2_new, nu_star, sigma2_old ) -
    log_post( sigma2_old, y, mu, nu_p, sigma2_p, n ) );
  if ( ratio > 1.0 ) return 1.0;
```

```
    else return ratio;
}

void main( )
/*
 *  Main routine
 */
{
  long int burnin, monitor, thin, n_run, i, n;
  double accept = 0.0, *y, sigma2_old, sigma2_star, nu_star, u,
    mu, nu_p, sigma2_p, sigma2_new, y_new, arg2, b;
  FILE *fp,*fpout;

  fp = fopen( "hastings.gaussian.variance.in", "r" );
  fscanf(fp,"%lf",&mu);
  fscanf(fp,"%lf",&nu_p);
  fscanf(fp,"%lf",&sigma2_p);
  fscanf(fp,"%lf",&sigma2_old);
  fscanf(fp,"%lf",&nu_star);
  fscanf(fp,"%ld",&burnin);
  fscanf(fp,"%ld",&monitor);
  fscanf(fp,"%ld",&thin);
  fscanf(fp,"%ld",&seed1);
  fscanf(fp,"%ld",&n);
  y = (double *)calloc(n,sizeof(double));
  for(i=0;i<n;i++)
     fscanf(fp,"%lf",&y[i]);
  fclose(fp);

  seed2 = 07;
  seed3 = 1973;

  fpout = fopen( "hastings.gaussian.variance.d", "w" );

  n_run = burnin + monitor;

  for ( i = 1; i <= n_run; i++ ) {
    arg2 = ( nu_star - 2.0 ) * sigma2_old / nu_star;
    sigma2_star = PD_sim( nu_star, arg2 );
    u = wichmann( );
    if ( u <= alpha( sigma2_old, sigma2_star, y, mu, nu_p,
      sigma2_p, nu_star, n ) ) b = 1.0;
    else
      b = 0.0;
    sigma2_new = sigma2_star * b + sigma2_old * ( 1.0 - b );
    arg2 = sqrt( sigma2_new );
    y_new = rnormal( mu, arg2 );
    if ( i > burnin ) accept = accept + b;
    if ( ( i > burnin ) && ( (i - burnin) % thin == 0 ) )
      fprintf( fpout, "%d   %10.4f   %10.4f\n", ( i - burnin ) /
        thin, sigma2_new, y_new );
    sigma2_old = sigma2_new;
  }
  printf( "%f\n", accept / monitor );
```

```
}
```

4 A Metropolis sampler in S+ for the Gaussian example in Section 2.2

```
  #
  # S+ functions to do Metropolis sampling in the model (2.7):
  #
  #   sigma2 ~ SI-chisq( nu.p, sigma2.p )
  #   ( y_i | sigma2 ) ~IID N( mu, sigma2 ), i = 1, ..., n
  #
  # Inputs:
  #
  #   y = data vector, of length n = sample size
  #   mu = known mean in Gaussian likelihood
  #   nu.p = prior effective sample size
  #   sigma2.p = prior estimate of sigma2
  #   sigma2.0 = initial value for sigma2 in Metropolis iterations
  #   kappa = scaling factor for Metropolis proposal distribution
  #     (affects the acceptance rate R; to increase R, decrease
  #     kappa)
  #   n.burnin = length of burn-in period
  #   n.monitor = length of monitoring period
  #   n.thin = thinning constant (only every n.thin-th iteration
  #     in the monitoring period will be written to disk)
  #   seed = random number seed (for generating repeatable
  #     sequences of Hastings iterations); must be an integer
  #     from 0 to 1000
  #   output.file.prefix = character string naming where you want
  #     the MCMC data set to go; for example, output.file.prefix
  #     = "NB10" would write the MCMC data set to the file
  #     "NB10.d"
  #
  # Outputs:
  #
  #   Acceptance rate R returned when iterations are finished
  #   A file called paste( output.file.prefix, ".d", sep = "" )
  #     is written (in the same directory where S+ has been
  #     called) containing one row for each monitored iteration
  #     and four columns: the monitored iteration number (from 1
  #     to n.monitor / n.thin), the simulated draws from the
  #     posterior for lambda = log(sigma2) and sigma2 for that
  #     iteration, and the corresponding simulated draw from the
  #     predictive distribution for a new y.star. If the output
  #     file exists before the function is invoked, it will be
  #     over-written

metropolis.gaussian.variance <- function( y, mu, nu.p, sigma2.p,
  sigma2.0, kappa, n.burnin, n.monitor, n.thin, seed,
  output.file.prefix ) {

  # Main routine
```

```
  lambda.old <- log( sigma2.0 )
  R <- 0
  write( c( lambda.old, R ), "loop.result", append = F )

  set.seed( seed )

  for ( i in 1:( n.burnin + n.monitor ) ) {

    null <- loop( y, mu, nu.p, sigma2.p, kappa,
      output.file.prefix, i )

  }

  loop.result <- scan( "loop.result" )
  R <- loop.result[2]
  return( R / n.monitor )

}

loop <- function( y, mu, nu.p, sigma2.p, kappa,
  output.file.prefix, i ) {

  loop.result <- scan( "loop.result" )
  lambda.old <- loop.result[1]
  R <- loop.result[2]
  n <- length( y )

  lambda.star <- PD.sim( lambda.old, kappa, n )
  u <- runif( 1 )
  b <- ( u <= alpha( lambda.old, lambda.star, y, mu, nu.p,
    sigma2.p ) )
  lambda.new <- lambda.star * b + lambda.old * ( 1 - b )
  y.new <- rnorm( 1, mu, sqrt( exp( lambda.new ) ) )

  if ( ( i > n.burnin ) ) R <- R + b
  if ( ( i > n.burnin ) & ( ( i - n.burnin ) %% n.thin == 0 ) )
    write( c( ( i - n.burnin ) / n.thin, signif( c( lambda.new,
      exp( lambda.new ), y.new ), digits = 5 ) ), paste(
      output.file.prefix, ".d", sep = "" ), ncol = 4, append =
      ( i > n.burnin + n.thin ) )

  lambda.old <- lambda.new
  write( c( lambda.old, R ), "loop.result", append = F )
  return( NULL )

}

PD.sim <- function( lambda, kappa, n ) {

  # Proposal distribution simulation

  return( rnorm( 1, lambda, sqrt( 2.0 * kappa / n ) ) )
```

```
}

alpha <- function( lambda.old, lambda.new, y, mu, nu.p,
  sigma2.p ) {

  # Acceptance probability calculation

  return( min( 1, exp( log.post( lambda.new, y, mu, nu.p,
    sigma2.p ) - log.post( lambda.old, y, mu, nu.p,
    sigma2.p ) ) ) )

}

log.post <- function( lambda, y, mu, nu.p, sigma2.p ) {

  # log( posterior ) calculation

  return( log.lik( lambda, y, mu ) + log.prior( lambda, nu.p,
    sigma2.p ) )

}

log.lik <- function( lambda, y, mu ) {

  # log( likelihood ) calculation

  n <- length( y )
  sigma2 <- exp( lambda )
  return( ( - n / 2 ) * log( sigma2 ) - sum( ( y - mu )^2 ) /
    ( 2 * sigma2 ) )

}

log.prior <- function( lambda , nu.p, sigma2.p ) {

  # log( prior ) calculation (including Jacobian)

  return( ( - nu.p / 2 ) * lambda - nu.p * sigma2.p /
    ( 2 * exp( lambda ) ) )

}
```

5 A Gibbs sampler in S+ for the uniform example in Section 2.5

```
  # S+ functions to do Gibbs sampling in the model (2.31, 2.32):
  #
  #   mu ~ U( L, H )
  #   ( sigma | mu ) ~ U( 0, min( mu - L, H - mu ) )
  #   ( y_i | mu, sigma ) ~IID U( mu - sigma, mu + sigma ),
  #     i = 1, ..., n
  #
  # (written by DD)
```

```
  #
  # Inputs:
  #
  #   y = data vector, of length n = sample size
  #   L = known lower bound for true range
  #   H = known upper bound for true range
  #   sigma.0 = initial value for sigma in Gibbs iterations
  #   n.burnin = length of burn-in period
  #   n.monitor = length of monitoring period
  #   n.thin = thinning constant (only every n.thin-th iteration
  #     in the monitoring period will be written to disk)
  #   seed = random number seed (for generating repeatable
  #     sequences of Gibbs iterations); must be an integer from 0
  #     to 1000
  #   output.file.prefix = character string naming where you want
  #     the MCMC data set to go; for example, output.file.prefix
  #     = "ammonite" would write the MCMC data set to the file
  #     "ammonite.d"
  #
  # Outputs:
  #
  #   A file called paste( output.file.prefix, ".d", sep = "" ) is
  #     written (in the same directory where S+ has been called)
  #     containing one row for each monitored iteration and five
  #     columns: the monitored iteration number (from 1 to
  #     n.monitor / n.thin), the simulated draws from the marginal
  #     posteriors for mu and sigma for that iteration, and the
  #     corresponding simulated draws from the marginal posteriors
  #     for A = mu - sigma and B = mu + sigma. If the output file
  #     exists before the function is invoked, it will be
  #     over-written

gibbs.uniform <- function( y, L, H, sigma.0, n.burnin, n.monitor,
  n.thin, seed, output.file.prefix ) {

  # Main routine

  sigma.previous <- sigma.0

  for ( i in 1:( n.burnin + n.monitor ) ) {

    mu.hat <- sample.mu( y, sigma.previous, L, H )
    sigma.hat <- sample.sigma( y, mu.hat, L, H )
    A.hat <- mu.hat - sigma.hat
    B.hat <- mu.hat + sigma.hat

  if ( ( i > n.burnin ) & ( ( i - n.burnin ) %% n.thin == 0 ) )
    write( c( ( i - n.burnin ) / n.thin, signif( c( mu.hat,
      sigma.hat, A.hat, B.hat ), digits = 6 ) ), paste(
      output.file.prefix, ".d", sep = "" ), ncol = 5, append =
      ( i > n.burnin + n.thin ) )

    sigma.previous <- sigma.hat
```

```
  }

  return( paste( n.monitor / n.thin,
    " iterations written to ", output.file.prefix, ".d",
    sep = "" ) )

}

sample.mu <- function( y, sigma, L, H ) {

  # Sampling from the full conditional for mu

  c.1 <- max( L + sigma, max( y ) - sigma )
  c.2 <- min( y ) + sigma
  c.3 <- min( H - sigma, c.2 )
  sigma.star <- ( H - L ) / 2.0
  c.4 <- 1.0 / log( sigma.star^2 / ( ( c.1 - L ) * ( H - c.3 ) ) )
  mu.star <- ( H + L ) / 2.0
  U <- runif( 1 )

  if ( sigma < mu.star - min( y ) ) {

    return( L + ( c.1 - L ) * ( ( c.2 - L ) / ( c.1 - L ) )^U )

  }

  else if ( U < c.4 * log( sigma.star / ( c.1 - L ) ) ) {

    return( L + ( c.1 - L ) * exp( U / c.4 ) )

  }

  else {

    return( H - sigma.star / exp( U / c.4 - log( sigma.star /
      ( c.1 - L ) ) ) )

  }

}

sample.sigma <- function( y, mu, L, H ) {

  # Sampling from the full conditional for sigma

  n <- length( y )
  c.5 <- max( mu - min( y ), max( y ) - mu )
  c.6 <- min( mu - L, H - mu )
  U <- runif( 1 )

  return( ( ( 1.0 - U ) * c.5^( 1.0 - n ) + U * c.6^( 1.0 - n ) )^
    ( 1.0 / ( 1.0 - n ) ) )

}
```

6 Computing approximate posterior covariance matrices in Maple, in the t model of Section 2.6

```
#
# Maple code to find the posterior mode and compute its
#   approximate covariance matrix (based on the Hessian at the
#   mode), in the NB10 $t$ model of Section 2.6.
#
# (written by DD, with some help from Riccardo Gatto)
#
# Input:
#
#   Reads from a file called "nb10.dat" which contains the NB10
#   measurements, 1 per line for 100 lines
#
# Outputs:
#
#   Obtains the MAP (maximum a posteriori) equations by
#   differentiating the log posterior function symbolically,
#   solves them numerically to get the mode, calculates the
#   Hessian symbolically, and evaluates it numerically at the mode
#

n := 100;

readlib( readdata );
y := readdata( `nb10.dat`, float, 1 );

mu.0 := 0.0;
sigma.mu := 500.0;
eta.0 := 3.80;
sigma.eta := 0.77;
lambda.0 := 1.84;
sigma.lambda := 0.59;

log.prior := -0.5 * ( ( mu - mu.0 ) / sigma.mu )^2 - 0.5 *       \
  ( ( eta - eta.0 ) / sigma.eta )^2 - 0.5 * ( ( lambda -         \
  lambda.0 ) / sigma.lambda )^2;

log.likelihood := n * lnGAMMA( 0.5 * ( exp( lambda ) +           \
  1.0 ) ) - n * eta - n * lnGAMMA( 0.5 * exp( lambda ) ) -       \
  0.5 * n * lambda - 0.5 * ( exp( lambda ) + 1.0 ) * sum(        \
  log( 1.0 + exp( - ( lambda + 2.0 * eta ) ) * ( y[i] -          \
  mu )^2 ), i = 1 .. n );

log.posterior := log.prior + log.likelihood;
map.eq1 := diff( log.posterior, mu );
map.eq2 := diff( log.posterior, eta );
map.eq3 := diff( log.posterior, lambda );

fsolve( { map.eq1, map.eq2, map.eq3 }, { mu, eta, lambda },      \
  { mu =  403 .. 406, eta = 1 .. 3, lambda = 0.5 .. 3.0 } );

# At this point Maple takes about 2.5 seconds at 333Mhz to
```

```
#   iteratively solve the MAP equations, obtaining the values of
#   mu, eta, and lambda listed below.

with( linalg );
H := hessian( log.posterior, [mu, eta, lambda] );

mu := 404.2956374;
eta := 1.346258072;
lambda := 1.259790967;

He := matrix( 3, 3, ( i,j ) -> 0 ):
for i from 1 to 3 do
  for j from 1 to 3 do
    He[i,j] := evalf( H[i,j] ):
  od:
od:

Sigma := inverse( - He );

# At this point Maple returns the covariance matrix:
#
#                [ .2159336246     .002989379323    .008083662383]
#                [                                              ]
#       Sigma := [.002989379323    .01193790730     .01486075432 ]
#                [                                              ]
#                [.008083662383    .01486075432     .07490518257 ]
```

7 A generic Metropolis sampler in S+, applied to the t model in Section 2.6

```
#
# S+ functions to do Metropolis sampling in the model (2.46,
#   2.48):
#
#   theta = ( mu, eta, lambda )
#   ( theta ) ~ N( mu.0, sigma.mu2 ) * N( eta.0, sigma.eta2 ) *
#     N( lambda.0, sigma.lambda2 )
#   ( y_i | theta ) ~IID t.exp( lambda ) ( mu, exp( 2 * eta ) ),
#     i = 1, ..., n
#
# Inputs:
#
#   y = data vector, of length n = sample size
#   mu.0 = prior mean for mu
#   sigma.mu = prior SD for mu
#   eta.0 = prior mean for eta
#   sigma.eta = prior SD for eta
#   lambda.0 = prior mean for lambda
#   sigma.lambda = prior SD for lambda
#   kappa = scaling factor for Metropolis proposal distribution
#     (affects the acceptance rate R; to increase R, decrease
#     kappa)
#   Sigma = proposal distribution covariance matrix
```

```
  #   theta.0 = initial value for ( mu, eta, lambda ) in
  #     Metropolis iterations
  #   n.burnin = length of burn-in period
  #   n.monitor = length of monitoring period
  #   n.thin = thinning constant (only every n.thin-th iteration
  #     in the monitoring period will be written to disk)
  #   seed = random number seed (for generating repeatable
  #     sequences of Hastings iterations); must be an integer
  #     from 0 to 1000
  #   output.file.prefix = character string naming where you want
  #     the MCMC data set to go; for example, output.file.prefix
  #     = "NB10" would write the MCMC data set to the file
  #     "NB10.d"
  #
  # Outputs:
  #
  #   Acceptance rate R returned when iterations are finished
  #   A file called paste( output.file.prefix, ".d", sep = "" ) is
  #     written (in the same directory where S+ has been called)
  #     containing one row for each monitored iteration and six
  #     columns: the monitored iteration number (from 1 to
  #     n.monitor/n.thin), the simulated draws from the posterior
  #     for theta = ( mu, eta, lambda ) for that iteration, and
  #     the corresponding simulated draws from the posterior for
  #     sigma = exp( eta ) and nu = exp( lambda ). If the output
  #     file exists before the function is invoked, it will be
  #     over-written

metropolis.t <- function( y, mu.0, sigma.mu, eta.0, sigma.eta,
  lambda.0, sigma.lambda, kappa, Sigma, theta.0, n.burnin,
  n.monitor, n.thin, seed, output.file.prefix ) {

  # Main routine

  theta.old <- theta.0
  p <- length( theta.old )
  R <- 0
  write( c( theta.old, R ), "loop.result", append = F )
  set.seed( seed )
  L <- t( chol( Sigma ) )
  L.kappa <- sqrt( kappa ) * L

  for ( i in 1:( n.burnin + n.monitor ) ) {

    null <- loop( p, L.kappa, y, mu.0, sigma.mu, eta.0, sigma.eta,
      lambda.0, sigma.lambda, output.file.prefix, i )

  }

  loop.result <- scan( "loop.result" )
  R <- loop.result[p + 1]
  return( R / n.monitor )

}
```

```
loop <- function( p, L.kappa, y, mu.0, sigma.mu, eta.0, sigma.eta,
  lambda.0, sigma.lambda, output.file.prefix, i ) {

  loop.result <- scan( "loop.result" )
  theta.old <- loop.result[1:p]
  R <- loop.result[p+1]
  n <- length( y )

  theta.star <- PD.sim( theta.old, p, L.kappa )
  u <- runif( 1 )
  b <- ( u <= alpha( theta.old, theta.star, y, mu.0, sigma.mu,
    eta.0, sigma.eta, lambda.0, sigma.lambda ) )
  theta.new <- theta.star * b + theta.old * ( 1 - b )

  if ( ( i > n.burnin ) ) R <- R + b
  if ( ( i > n.burnin ) & ( ( i - n.burnin ) %% n.thin == 0 ) )
    write( c( ( i - n.burnin ) / n.thin, signif( c( theta.new,
      exp( theta.new[c( 2, 3 )] ) ), digits = 5 ) ), paste(
      output.file.prefix, ".d", sep = "" ), ncol = p + 3,
      append = ( i > n.burnin + n.thin ) )

  theta.old <- theta.new
  write( c( theta.old, R ), "loop.result", append = F )
  return( NULL )

}

PD.sim <- function( theta, p, L.kappa ) {

  # Proposal distribution simulation

  Z <- matrix( rnorm( p ), p, 1 )
  Mu <- matrix( theta, p, 1 )
  theta.star <- c( Mu + ( L.kappa %*% Z ) )

  return( theta.star )

}

alpha <- function( theta.old, theta.new, y, mu.0, sigma.mu, eta.0,
  sigma.eta, lambda.0, sigma.lambda ) {

  # Acceptance probability calculation

  return( min( 1, exp( log.post( theta.new, y, mu.0, sigma.mu,
    eta.0, sigma.eta, lambda.0, sigma.lambda ) - log.post(
    theta.old, y, mu.0, sigma.mu, eta.0, sigma.eta, lambda.0,
    sigma.lambda ) ) ) )

}

log.post <- function( theta, y, mu.0, sigma.mu, eta.0, sigma.eta,
  lambda.0, sigma.lambda ) {
```

```
  # log( posterior ) calculation

  return( log.prior( theta, mu.0, sigma.mu, eta.0, sigma.eta,
    lambda.0, sigma.lambda ) + log.lik( theta, y ) )

}

log.prior <- function( theta, mu.0, sigma.mu, eta.0, sigma.eta,
  lambda.0, sigma.lambda ) {

  # log( prior ) calculation (including Jacobian)

  mu <- theta[1]
  eta <- theta[2]
  lambda <- theta[3]

  return( -0.5 * ( ( mu - mu.0 ) / sigma.mu )^2 - 0.5 *
    ( ( eta - eta.0 ) / sigma.eta )^2 - 0.5 * ( ( lambda -
    lambda.0 ) / sigma.lambda )^2 )

}

log.lik <- function( theta, y ) {

  # log( likelihood ) calculation

  mu <- theta[1]
  eta <- theta[2]
  lambda <- theta[3]
  n <- length( y )

  return( n * lgamma( 0.5 * ( exp( lambda ) + 1.0 ) ) - n * eta -
    n * lgamma( 0.5 * exp( lambda ) ) - 0.5 * n * lambda - 0.5 *
    ( exp( lambda ) + 1.0 ) * sum( log( 1.0 + exp( - ( lambda +
    2.0 * eta ) ) * ( y - mu )^2 ) ) )

}
```

8 BUGS files for Gibbs sampling in the t example of Section 2.6

Section 2.6 contains the file `nb10.3.bug` for fitting the t model (2.46, 2.48) to the NB10 data via BUGS. Other files used in running BUGS in this example are as follows.

NB10-y.dat

```
405
402
408
399
 .
 .    (100 rows)
 .
398
406
403
404
```

NB10-grid.dat

```
0.00105225
0.00169802
0.00250893
0.00346489
     .
     .        (101 rows)
     .
0.00257974
0.00249658
0.00241623
0.00233859
0.000977389
```

NB10.3.cmd

```
compile( "nb10.5.bug" )
update( 1000 )
monitor( mu, 14 )
monitor( sigma, 14 )
monitor( nu, 14 )
update( 70000 )
q( )
```

NB10.3.in

```
list( mu = 405.03, tau = 0.18233,
  u = 30 )
```

9 Metropolis and Gibbs sampling in multilevel models via MLwiN

References

Abramowitz M, Stegun IA (eds.; 1972). *Handbook of Mathematical Functions*. New York: Dover.

Anderson TW (1971). *Statistical Analysis of Time Series*. New York: Wiley.

Barnard GA (1958). Thomas Bayes—a biographical note (together with a reprinting of Bayes, 1763). *Biometrika*, **45**, 293–315. Reprinted in Pearson and Kendall (1970), 131–153.

Baum LE, Petrie T, Soules G, Weiss N (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, **41**, 164–171.

Bayes T (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, **53**, 370–418. Reprinted in Barnard (1958) and Pearson and Kendall (1970), 131–153.

Becker RA, Chambers JM, Wilks AR (1988). *The New S Language*. Pacific Grove CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Bernardo JM (1979). Reference posterior distributions for Bayesian inference (with discussion). *Journal of the Royal Statistical Society Series B*, **41**, 113–147.

Bernardo JM, Smith AFM (1994). *Bayesian Theory*. New York: Wiley.

Best NG, Cowles MK, Vines SK (1995). *CODA Manual version 0.30*. MRC Biostatistics Unit, Cambridge UK. Updated in *CODA version 0.4: manual addendum*, 1997.

Box GEP (1980). Sampling and Bayes' inference in scientific modeling (with discussion). *Journal of the Royal Statistical Society, Series A*, **143**, 383–430.

Box GEP, Jenkins GM (1976). *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.

Breiman L (1968). *Probability*. Reading MA: Addison-Wesley.

Brooks SP, Roberts GO (1995). Diagnosing convergence of Markov chain Monte Carlo algorithms. Technical Report 95–12, Statistical Laboratory, University of Cambridge UK.

Browne W (1998). *Computational Aspects of Multi-Level Modelling*. PhD dissertation, Department of Mathematical Sciences, University of Bath (in preparation).

Carlin BP, Louis TA (1996). *Bayes and Empirical Bayes Methods for Data Analysis*. London: Chapman & Hall.

Chatfield C (1996). *The Analysis of Time Series*, fifth edition. London: Chapman & Hall.

Cochran WG (1977). *Sampling Techniques*, third edition. New York: Wiley.

Cowles MK, Carlin BP (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, **91**, 883–904.

Craig PS, Goldstein M, Seheult AH, Smith JA (1997). Constructing partial prior specifications for models of complex physical systems. *The Statistician*, **46**, forthcoming.

Dempster AP, Laird NM, Rubin DB (1978). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society Series B*, **39**, 1–38.

Diaconis P, Freedman D (1980). Finite exchangeable sequences. *Annals of Probability*, **8**, 745–764.

Doss H, Narasimhan B (1994). Bayesian Poisson regression using the Gibbs sampler: Sensitivity analysis through dynamic graphics. Technical report, Department of Statistics, Ohio State University.

Draper D (1995a). Inference and hierarchical modeling in the social sciences (with discussion). *Journal of Educational and Behavioral Statistics*, **20**, 115–147, 233–239.

Draper D (1995b). Assessment and propagation of model uncertainty (with discussion). *Journal of the Royal Statistical Society Series B*, **57**, 45–97.

Draper D, Hodges JS, Mallows CL, Pregibon D (1993). Exchangeability and data analysis (with discussion). *Journal of the Royal Statistical Society, Series A*, **156**, 9–37.

Draper D (1997). On the relationship between model uncertainty and inferential/predictive uncertainty. Under revision for *Biometrika*.

Feller W (1968). *An Introduction to Probability Theory and Its Applications*, volume I, third edition (revised printing). New York: Wiley.

Feller W (1971). *An Introduction to Probability Theory and Its Applications*, volume II, second edition. New York: Wiley.

Ferguson TS (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, **1**, 209–230.

Ferguson TS (1974). Prior distributions on the space of probability measures. *Annals of Statistics*, **2**, 615–629.

de Finetti B (1930). Funzione caratteristica di un fenomeno aleatorio. *Mem. Acad. Naz. Lincei*, **4**, 86–133.

de Finetti B (1937/1980). La prévision: Ses lois logiques, ses sources subjectives. *Annales d'Institute de Henri Poincaré*, **7**, 1–68. Reprinted as "Foresight: its logical laws, its subjective sources," in *Studies in Subjective Probability*, HE Kyburg, Jr., HE Smokler, eds., New York: Wiley (1980), 93–158.

de Finetti B (1974/5). *Theory of Probability*, volumes 1 and 2. New York: Wiley.

Fisher RA (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London A*, **222**, 309–368.

Fisher RA (1956). *Statistical Methods and Scientific Inference*. London: Oliver and Boyd.

Freedman DA (1963). On the asymptotic behavior of Bayes estimates in the discrete case, I. *Annals of Mathematical Statistics*, **34**, 1386–1403.

Freedman DA (1995). Some issues in the foundations of statistics (with discussion). *Foundations of Science*, **1**, 1–83.

Freedman D, Pisani R, Purves R (1998). *Statistics*, third edition. New York: Norton.

Gamerman D (1997). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. New York: Chapman & Hall.

Gelfand AE, Smith AFM (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85**, 398–409.

Gelman A, Carlin JB, Stern HS, Rubin DB (1995). *Bayesian Data Analysis*. London: Chapman & Hall.

Gelman A, Roberts GO, Gilks WR (1996). Efficient Metropolis jumping rules. In *Bayesian Statistics 5*, Bernardo JM, Berger JO, Dawid AP, Smith AFM (eds.), 599–607. Oxford: Clarendon Press.

Gelman A, Rubin DB (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, **7**, 457–511.

Geman S, Geman D (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721–741.

Geweke J (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, **57**, 1317–1339.

Geweke J (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments (with discussion). In *Bayesian Statistics 4*, Bernardo JM, Berger JO, Dawid AP, Smith AFM (eds.), 169–193. Oxford: Clarendon Press.

Gilks WR (1992). Derivative-free adaptive rejection sampling for Gibbs sampling. In *Bayesian Statistics 4*, Bernardo JM, Berger JO, Dawid AP, Smith AFM (eds.), 641–649. Oxford: Clarendon Press.

Gilks WR, Richardson S, Spiegelhalter DJ (1996a). *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall.

Gilks WR, Richardson S, Spiegelhalter DJ (1996b). Introducing Markov Chain Monte Carlo. In *Markov Chain Monte Carlo in Practice*, Gilks WR, Richardson S, Spiegelhalter DJ (eds.), 1–19. London: Chapman & Hall.

Gilks WR, Roberts GO, Sahu SK (1997). Adaptive Markov Chain Monte Carlo through regeneration. Technical report, MRC Biostatistics Unit, Cambridge UK.

Gilks WR, Thomas A, Spiegelhalter DJ (1994). A language and programming for complex Bayesian modelling. *The Statistician*, **43**, 169–178.

Gilks WR, Wild P (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, **41**, 337–348.

Hacking I (1975). *The Emergence of Probability*. Cambridge: Cambridge University Press.

Hastings WK (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Heidelberger P, Welch P (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, **31**, 1109–1144.

Johnson NL, Kotz S (1970). *Distributions in Statistics: Continuous Univariate Distributions*, volume 1. New York: Wiley.

Kadane JB, Dickey JM, Winkler RL, Smith WS, Peters SC (1980). Interactive elicitation of opinion for a normal linear model. *Journal of the American Statistical Association*, **75**, 845–854.

Kadane JB, Wolfson LJ (1997). Experiences in elicitation. *The Statistician*, **46**, forthcoming.

Kahn K, Rubenstein L, Draper D, Kosecoff J, Rogers W, Keeler E, Brook R (1990). The effects of the DRG-based Prospective Payment System on quality of care for hospitalized Medicare patients: An introduction to the series. *Journal of the American Medical Association*, **264**, 1953–1955 (with editorial comment, 1995–1997).

Keeler E, Kahn K, Draper D, Sherwood M, Rubenstein L, Reinisch E, Kosecoff J, Brook R (1990). Changes in sickness at admission following the introduction of the Prospective Payment System. *Journal of the American Medical Association*, **264**, 1962–1968.

Knuth D (1984). *The TEXbook*. Reading MA: Addison-Wesley.

Ku HH (ed.; 1969). *Precision Measurement and Calibration*. National Bureau of Standards Special Publication number 300, volume 1 (Washington DC).

Lamport L (1994). *LATEX: A Document Preparation System*, second edition. Reading MA: Addison-Wesley.

Laplace PS (1774). Mémoire sur la probabilité des causes par les évenements. *Mémoires de l'Academie des Sciences de Paris*, **6**, 621–656. English translation in 1986 as "Memoir on the probability of the causes of events," with an introduction by SM Stigler, *Statistical Science*, **1**, 359–378.

Lauritzen SL, Dawid AP, Larsen BN, Leimer H-G (1990). Independence properties of directed Markov fields. *Networks*, **20**, 491–505.

LeCam L, Yang EL (1990). *Asymptotics in Statistics: Some Basic Concepts*. New York: Springer-Verlag.

Lehmann EL (1983). *Theory of Point Estimation*. New York: Wiley.

Lindley DV (1972). *Bayesian Statistics: A Review*. Philadelphia PA: SIAM.

Macellari CE (1986). Late Campanian-Maastrichtian ammonite fauna from Seymour Island (Antarctic peninsula). *Journal of Paleontology*, **60**, 1–55.

Madigan D, Raftery AE (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, **89**, 1535–1546.

Mallows CL (1998). The zeroth problem. Technical report, AT&T Bell Labs, Murray Hill NJ (1998 Fisher Lecture, Joint Statistical Meetings, Anaheim CA).

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1092.

Mosteller F, Wallace DL (1964). *Inference and Disputed Authorship: The Federalist*. Reading MA: Addison-Wesley. Second edition published (1984) as *Applied Bayesian and Classical Inference: The Case of* The Federalist *Papers*. New York: Springer-Verlag.

von Neumann J, Morgenstern O (1944). *Theory of Games and Economic Behavior*. Princeton: Princeton University Press.

Oakes M (1990). *Statistical Inference*. Chestnut Hill MA: Epidemiology Resources.

O'Hagan A (1997). Eliciting expert beliefs in substantial practical applications. *The Statistician*, **46**, forthcoming.

Pearson ES, Kendall MG (eds.) (1970). *Studies in the History of Statistics and Probability*. London: Charles Griffin.

Raftery AL, Lewis S (1992). How many iterations in the Gibbs sampler? In *Bayesian Statistics 4*, Bernardo JM, Berger JO, Dawid AP, Smith AFM (eds.), 763–774. Oxford: Clarendon Press.

Ramsay FP (1931/1980). Truth and probability. In *The Foundations of Mathematics and Other Logical Essays*, RB Braithwaite, ed., London: Kegan, Paul, Trench, & Trubner, 156–198. Reprinted in *Studies in Subjective Probability*, HE Kyburg, Jr., and HE Smokler, eds., New York: Wiley (1980), 61–92.

Ripley BD (1987). *Stochastic Simulation*. New York: Wiley.

Roberts GO (1996). Markov chain concepts related to sampling algorithms. In *Markov Chain Monte Carlo in Practice*, Gilks WR, Richardson S, Spiegelhalter DJ (eds.), 45–57. London: Chapman & Hall.

Ross SM (1970). *Applied Probability Models with Optimization Applications*. San Francisco: Holden-Day.

Rubin DB (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics*, **12**, 1151–1172.

Rubin DB (1987). A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing data are modest: the SIR algorithm. Discussion of Tanner and Wong (1987).

Samaniego FJ, Reneau DM (1994). Toward a reconciliation of the Bayesian and frequentist approaches to point estimation. *Journal of the American Statistical Association*, **89**, 947–957.

Savage LJ (1954). *The Foundations of Statistics*. New York: Wiley.

Shapiro SS, Wilk M (1965). An analysis of variance test for normality (complete samples). *Biometrika*, **52**, 591–611.

Silverman BW (1986). *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

Smith AFM, Skene AM, Shaw JEH, Naylor JC, Dransfield M (1985). The implementation of the Bayesian paradigm. *Communications in Statistics, Theory and Methods*, **14**, 1079–1109.

Snedecor GW, Cochran WG (1980). *Statistical Methods*, seventh edition. Ames IA: Iowa State University Press.

Spiegelhalter DJ, Thomas A, Best NG, Wilks WR (1995). *BUGS: Bayesian Analysis Using Gibbs Sampling, version 0.50*. MRC Biostatistics Unit, Cambridge UK. Updated in *BUGS 0.6: Bayesian Analysis Using Gibbs Sampling (Addendum to Manual)*, 1997.

Stander J, Silverman BW (1994). [xx supply this]

Stigler SM (1986). *The History of Statistics: The Measurement of Uncertainty before 1900*. Cambridge MA: Harvard University Press.

Strauss D, Sadler PM (1989). Classical confidence intervals and Bayesian probability estimates for ends of local taxon ranges. *Mathematical Geology*, **21**, 411–427.

Tanner MA, Wong WH (1987). The calculation of posterior distributions by data augmentation (with discussion). *Journal of the American Statistical Association*, **82**, 528–550.

Tierney L (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, **22**, 1701–1762.

Tierney L (1996). Introduction to general state-space Markov chain theory. In *Markov Chain Monte Carlo in Practice*, Gilks WR, Richardson S, Spiegelhalter DJ (eds.), 59–74. London: Chapman & Hall.

Tierney L, Kadane JB (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, **81**, 82–86.

Venables WN, Ripley BD (1997). *Modern Applied Statistics with* S+, second edition. New York: Springer-Verlag.

Walker SG, Damien P, Laud PW, Smith AFM (1997). Bayesian nonparametric inference for random distributions and related functions. Technical report, Department of Mathematics, Imperial College, London.

Whittaker J (1990). *Graphical Models in Applied Multivariate Analysis*. New York: Wiley.

Index