PART I

Introduction to hierarchical modeling

As background for this volume, we begin with the general framework for hierarchical Bayes in Chapter 1, followed by an application to population genetics in Chapter 2, where the hierarchical structure is especially transparent. In Chapter 1, Carlin et al. start with Bayes' theorem and demonstrate how one moves to hierarchical Bayes models, where the flexibility of Bayes becomes immediately apparent. They use graphs to illustrate this structure and suggest that a substructure based on "data models," "process models," and "parameter models" can aid model conceptualization and development. They further outline application of MCMC for analysis. All of the chapters in this volume can be viewed from the perspective of this basic framework.

Holsinger's application in Chapter 2 provides a clear illustration of the approach. The "data model" in this context is the "uncertainty associated with *statistical sampling*" or the binomial/multinomial likelihood of obtaining the observed number of alleles in a sample. The "process model" in this context is the *genetic sampling*, the inherent stochasticity in the evolutionary process. The parameter model takes up the fact that allelic frequencies are organized in terms of populations, where there can be variability among populations.

CHAPTER 1

Elements of hierarchical Bayesian inference

Bradley P. Carlin, James S. Clark, and Alan E. Gelfand

Serious investigation of ecological processes is challenging due to the complex nature of these processes and the lack of sufficient data to *see* them well. Hence, acknowledging our limitations, we turn to stochastic modeling as a means to capture the uncertainty in our inference about the process. Since typically, such processes involve components at different levels, stages, and scales, it is natural to frame our modeling in the context of hierarchical models. In turn, since such models introduce unknowns, for example, parameters or latent processes, we need to incorporate the uncertainty associated with these unknowns in order to achieve a better overall assessment of the uncertainty in our modeling. This encourages us to cast the models under the Bayesian framework.

The objective of this first chapter is to provide an introduction to the tools that we need to work with Bayesian hierarchical modeling. Indeed, we acknowledge that the required material here is substantial, that we are only opening the proverbial Pandora's box, and that readers wishing to attempt the use of this technology will have to invest a significant amount of time to climb the learning curve, to achieve comfort with the technology. To that end, we have supplied a rich reference list. And, we emphasize that the reward is substantial. The ability to let the problem and, more generally, the science drive the modeling rather than forcing the analysis to fit a standard technique, to enable a canned package is wonderfully liberating and considerably enhances the way one approaches the understanding of complex systems.

The tools we discuss here include the Bayesian inference paradigm, hierarchical modeling and, more generally, directed graphical models, model comparison, and Bayesian computation, that is, computation to fit Bayesian hierarchical models. Though the development of each is, of necessity brief, we do provide exemplification and as noted above, direction to further resources.

1.1 The challenge of ecological modeling

Complex interrelationships combined with poor visibility make environmental modeling hard. Rarely can the environmental scientist hope to isolate one or a few variables to meet assumptions of classical statistical models. We might even question whether such abstraction is desirable. Relevant ecological and evolutionary processes play out in heterogeneous landscapes, where context varies widely in space and time.

Is it reasonable to extrapolate results from a highly abstracted experimental system with limited spatial extent and duration to natural and managed ecosystems? There are reasons to question extrapolation and to attempt inference and prediction directly from data obtained at the relevant scales. Here we confront not only the complexity of interacting processes, which must be recognized if we are to quantify the important relationships, but also the obscure relationships that impinge between the processes we care about and the data that can be had. Traditional methods have trouble with large numbers of uncontrolled variables (each must involve a stochastic component). Rarely can we observe the processes directly; rather we derive clues from information that is indirect and that arrives from nonuniform

methods and uneven sampling effort. We may have many types of information that are not independent of one another, yet together they should provide a richer understanding than if each were taken in isolation.

Hierarchical Bayes provides new tools for drawing inference, for prediction, and for decision making. As demonstrated by chapters in this book, the promise for environmental sciences is large. In this chapter we lay out the basic tools that arise in the chapters that follow. We introduce the elements of hierarchical Bayes and summarize analysis. Rather than provide specific examples, we cross-reference where methods we introduce are applied in remaining book chapters. You will find here a common set of tools applied to such disparate topics as population genetics (Chapter 2), experimental and monitoring studies that involve time series (Chapter 5), spatial and spatio-temporal treatment of populations (Chapter 4), communities (Chapter 3), ecosystems (Chapter 6), and the atmosphere (Chapter 7). Each of these approaches builds on the hierarchical Bayes framework that involves "models" for context, process, and data, and exploits simple, conditional relationships as the basic modeling unit. As different as these studies are, you will recognize a common construction and strategy for analysis. By providing a range of such applications, we hope to both emphasize the potential and provide examples. We begin with a brief introduction of the Bayesian model and the principles of Bayesian inference. We then move to hierarchical structures, where the Bayesian model is extended to high dimensional problems that can be viewed as a network, most of which is invisible (and must be inferred). Finally, we turn to analysis, laying out the principles of the Gibbs sampling framework, and how we use it for inference and prediction.

Currently, many good Bayesian books are available, and we list a few of them and their characteristics. First we must mention the texts stressing Bayesian theory, including DeGroot (1970), Berger (1985), Bernardo and Smith (1994), and Robert (1994). These books tend to focus on foundations and decision theory, rather than computation or data analysis. On the more methodological side, nice introductory books are those of Lee (1997) and Congdon (2001). The books by Carlin and Louis (2000), by Gelman et al. (2004), and by O'Hagan (1994) offer more general Bayesian modeling treatments. Clark (2005) covers both classical and Bayesian frameworks with specific applications to ecology.

1.2 Introduction to hierarchical modeling and Bayes' Theorem

By modeling both the observed data and any unknowns as random variables, the Bayesian approach to statistical analysis provides a cohesive framework for combining complex data models and external knowledge or expert opinion. In addition to specifying the distributional model $f(\mathbf{y}|\boldsymbol{\theta})$ for the observed data $\mathbf{y} = (y_1, \ldots, y_n)$ given a vector of unknown parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$, we suppose that $\boldsymbol{\theta}$ is a random quantity sampled from a *prior* distribution $\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ is a vector of hyperparameters. For instance, y_i might be the observed abundance of a particular species in areal unit i or it might be the observed frequency of a particular allele type in population i. θ_i would then be true abundance of the species in unit i or the true allele proportion in population i. Finally, $\boldsymbol{\lambda}$ is a parameter controlling, say, spatial similarity across areal units or, say, variation among populations. If $\boldsymbol{\lambda}$ is known, inference concerning $\boldsymbol{\theta}$ is based on its *posterior* distribution,

$$p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\lambda}) = \frac{p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\lambda})}{p(\mathbf{y}|\boldsymbol{\lambda})} = \frac{p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\lambda})}{\int p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\lambda}) d\boldsymbol{\theta}}$$
$$= \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})}{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda}) d\boldsymbol{\theta}}. \quad (1.1)$$

Notice the contribution of both the data (in the form of the likelihood f) and the external knowledge or opinion (in the form of the prior π) to the posterior. Since, in practice, $\boldsymbol{\lambda}$ will not be known, a second stage (or *hyperprior*) distribution $h(\boldsymbol{\lambda})$ will often be required, and (1.1) will be replaced with

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}, \boldsymbol{\theta})}{p(\mathbf{y})} = \frac{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})h(\boldsymbol{\lambda}) d\boldsymbol{\lambda}}{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})h(\boldsymbol{\lambda}) d\boldsymbol{\theta} d\boldsymbol{\lambda}}.$$

The Bayesian inferential paradigm offers potentially attractive advantages over the classical, frequentist statistical approach through its more

philosophically sound foundation, its unified approach to data analysis, and its ability to formally incorporate prior opinion or external empirical evidence into the results via the prior distribution π. The Bayesian approach better captures uncertainty in our inference (both with regard to parameters in models and with regard to model specification itself). It also provides exact inference, avoiding asymptotics whose adequacy may be difficult to assess. Indeed, such asymptotics may be inappropriate in complex models. Scientists, formerly reluctant to adopt the Bayesian approach due to its *subjectivity* and a lack of necessary computational tools, are now turning to it with increasing regularity as classical methods emerge as both theoretically and practically inadequate. Modeling the θ_i as random (instead of fixed) effects allows us to induce specific (e.g. spatial) correlation structures among them, hence among the observed data y_i as well. As an aside, in (1.1) we might replace $\boldsymbol{\lambda}$ by an estimate $\hat{\boldsymbol{\lambda}}$ being the value of *blam* that maximizes the marginal distribution $p(\mathbf{y}|\boldsymbol{\lambda}) = \int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})d\boldsymbol{\theta}$, viewed as a function of $\boldsymbol{\lambda}$. Inference could then proceed based on the *estimated* posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}, \hat{\boldsymbol{\lambda}})$, obtained by plugging $\hat{\boldsymbol{\lambda}}$ into equation (1.1). This approach is referred to as *empirical Bayes* analysis; see Berger (1985), Maritz and Lwin (1989), and Carlin and Louis (2000) for details regarding empirical Bayes methodology and applications.

Hierarchical Bayesian methods now enjoy broad scientific application with increasing application in ecology, evolutionary biology and climatology, as the remainder of this book reveals. A computational challenge in applying Bayesian methods comes from the fact that, for most realistic problems, the integrations required to do inference under (1.1) are generally not tractable in closed form, and thus must be approximated numerically. Forms for π and h (called *conjugate* priors) that enable at least partial analytic evaluation of these integrals may often be found, but in hierarchical models of interest, intractable integrations will remain. Here the emergence of inexpensive, high-speed computing equipment and software comes to the rescue, enabling the application of recently developed Markov chain Monte Carlo (MCMC) integration methods, such as the Metropolis–Hastings algorithm (Metropolis et al. 1953; Hastings 1970) and the Gibbs sampler (Geman and Geman 1984; Gelfand and Smith 1990).

1.2.1 Illustrations of Bayes' Theorem

Equation (1.1) is referred to as *Bayes' Theorem* or *Bayes' Rule*. We illustrate its use with two standard normally distributed data examples.

Suppose we have observed a single normal (Gaussian) observation $Y \sim N(\theta, \sigma^2)$ with σ^2 known, so that the likelihood $f(y|\theta) = N(y|\theta, \sigma^2) \equiv 1/(\sigma\sqrt{2\pi})\exp(-(y-\theta)^2/2\sigma^2), y \in \Re, \theta \in \Re$, and $\sigma > 0$. If we specify the prior distribution as $\pi(\theta) = N(y|\mu, \tau^2)$ with $\boldsymbol{\lambda} = (\mu, \tau^2)'$ fixed, then from (1.1) we can compute the posterior as

$$\begin{aligned} p(\theta|y) &= \frac{N(\theta|\mu, \tau^2)N(y|\theta, \sigma^2)}{p(y)} \\ &\propto N(\theta|\mu, \tau^2)N(y|\theta, \sigma^2) \\ &= N\Bigg(\theta \,\Big|\, \frac{\sigma^2}{\sigma^2+\tau^2}\mu \\ &\qquad + \frac{\tau^2}{\sigma^2+\tau^2}y, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\Bigg). \end{aligned} \tag{1.2}$$

That is, the posterior distribution of θ given y is also normal with mean and variance as given. The proportionality in the second row arises since the marginal distribution $p(y)$ does not depend on θ, and is thus constant with respect to the Bayes' Theorem calculation. The final equality in the third row results from collecting like (θ^2 and θ) terms in the exponential and then completing the square.

Note that the posterior mean $E(\theta|y)$ is a weighted average of the prior mean μ and the data value y, with the weights depending on our relative uncertainty with respect to the prior and the likelihood. Also, the posterior *precision* (reciprocal of the variance) is equal to $1/\sigma^2 + 1/\tau^2$, which is the sum of the likelihood and prior precisions. Thus, thinking of precision as "information," we see that in the normal/normal model, the information in the posterior is the total of the information in the prior and the likelihood.

Next, let us suppose, that instead of a single datum we have a set of n observations $\mathbf{y} = (y_1, y_2, \ldots, y_n)'$. From basic normal theory we know that $f(\bar{y}|\theta) = N(\theta, \sigma^2/n)$. Since $\bar{y}$ is sufficient for θ,

we have

$$\begin{aligned} p(\theta|\mathbf{y}) &= p\left(\theta|\bar{y}\right) \\ &= N\Bigg(\theta \,\Big|\, \frac{(\sigma^2/n)}{(\sigma^2/n)+\tau^2}\mu \\ &\quad + \frac{\tau^2}{(\sigma^2/n)+\tau^2}\bar{y}, \frac{(\sigma^2/n)\tau^2}{(\sigma^2/n)+\tau^2}\Bigg) \\ &= N\Bigg(\theta \,\Big|\, \frac{\sigma^2}{\sigma^2+n\tau^2}\mu \\ &\quad + \frac{n\tau^2}{\sigma^2+n\tau^2}\bar{y}, \frac{\sigma^2\tau^2}{\sigma^2+n\tau^2}\Bigg). \end{aligned}$$

which is a weighted average of the prior (μ) and data-supported ($\bar{y}$) values.

In these two examples, the prior leads to a posterior distribution for θ that is available in closed form, and it is a member of the same distributional family as the prior. Such a prior is referred to as a *conjugate* prior. Such priors are often used, because, when available, conjugate families are convenient and still allow a variety of shapes wide enough to capture our prior beliefs. Note that setting $\tau^2 = \infty$ in the previous example corresponds to a prior that is arbitrarily vague, or *noninformative*. This then leads to a posterior of $p(\theta|y) = N(\theta|\bar{y}, \sigma^2/n)$, exactly the same as the likelihood for this problem. This arises since the limit of the conjugate (normal) prior here is actually a uniform, or "flat" prior, and thus the posterior is nothing but the likelihood (possibly renormalized to integrate to 1 as a function of θ). Of course, the flat prior is *improper* here, since the uniform does not integrate to anything finite over the entire real line; however, the posterior is still well defined since the likelihood can be integrated with respect to θ.

More generally, let $\mathbf{Y}$ be an $n \times 1$ data vector, X an $n \times p$ matrix of covariates, and adopt the likelihood and prior structure,

$$\begin{aligned} \mathbf{Y}|\boldsymbol{\beta} &\sim N_n(X\boldsymbol{\beta}, \Sigma), \\ &\quad \text{that is, } f(\mathbf{Y}|\boldsymbol{\beta}) \equiv N_n(\mathbf{Y}|X\boldsymbol{\beta}, \Sigma), \\ \boldsymbol{\beta} &\sim N_p(A\boldsymbol{\alpha}, V), \\ &\quad \text{that is, } \pi(\boldsymbol{\beta}) \equiv N(\boldsymbol{\beta}|A\boldsymbol{\alpha}, V). \end{aligned}$$

Here $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients and Σ is a $p \times p$ covariance matrix. Then, as first discussed in Lindley and Smith (1972), the marginal distribution of $\mathbf{Y}$ is

$$\mathbf{Y} \sim N(XA\boldsymbol{\alpha}, \Sigma + XVX^T),$$

the posterior distribution of $\boldsymbol{\beta}|\mathbf{Y}$ is

$$\boldsymbol{\beta}|Y \sim N(D\mathbf{d}, D),$$

where

$$D^{-1} = X^T\Sigma^{-1}X + V^{-1}$$

and

$$\mathbf{d} = X^T\Sigma^{-1}\mathbf{Y} + V^{-1}A\boldsymbol{\alpha}.$$

Thus $E(\boldsymbol{\beta}|\mathbf{Y}) = D\mathbf{d}$ provides a point estimate for $\boldsymbol{\beta}$, with variability captured by the associated variance matrix D. In particular, note that for a vague prior we may set $V^{-1} = 0$, so that $D^{-1} = X\Sigma^{-1}X$ and $\mathbf{d} = X^T\Sigma^{-1}\mathbf{Y}$. In the simple case where $\Sigma = \sigma^2 I_p$, the posterior becomes

$$\boldsymbol{\beta}|Y \sim N(\hat{\boldsymbol{\beta}}, \sigma^2(X'X)^{-1}),$$

where $\hat{\boldsymbol{\beta}} = (X'X)^{-1}X'\mathbf{y}$. Since the usual likelihood approach produces

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(X'X)^{-1}),$$

we once again we see "flat prior" Bayesian results that are formally equivalent to the usual likelihood approach. Indeed, Bayesians often attempt to use flat or otherwise improper noninformative priors, since prior feelings are often rather vague relative to the information in the likelihood, and in any case we typically want the data (and not the prior) to dominate the determination of the posterior. However, when using improper priors, care must be taken to ensure that the resulting posterior is still proper, that is, that the product of the likelihood times the prior is still integrable with regard to all of the model parameters. This can be very demanding to check with complex multilevel models such as those we encounter in this book. Since we usually have some prior information on the magnitude of a parameter, we encourage the use of somewhat informative priors, reflecting rough knowledge of a center and a scale. Typically, this tends to provide better-behaved Bayesian computation.

AQ: Please check Lindley and Smith 1972 is not in ref. list.

1.3 Bayesian inference

While the computing associated with Bayesian methods can be daunting, the subsequent inference is relatively straightforward, especially in the case of estimation. Once we have computed (or obtained an estimate of) the posterior, inference comes down merely to summarizing this distribution. In other words, by Bayes' Rule the posterior summarizes everything we know about the model parameters in the light of the data. As we noted in the previous section, for hierarchical models, calculation of the posterior distribution of, for example, components of $\boldsymbol{\theta}$ or functions of $\boldsymbol{\theta}$ can not be done explicitly. Fortunately, the aforementioned simulation methods enable samples from these posterior distributions which, in turn, enables us to provide estimates of the distributions. However, in the remainder of this section, we shall assume for simplicity that the posterior $p(\boldsymbol{\theta}|\mathbf{y})$ itself is available for summarization. Bayesian methods for estimation are reminiscent of corresponding maximum likelihood methods. This should not be surprising, since likelihoods form an important part of the Bayesian calculation; we have even seen that a normalized (i.e. standardized) likelihood can be thought of a posterior when this is possible. Even with hierarchical models, associated Bayesian computation is analogous to EM algorithm methods. See, again, the book by Gelman et al. (2004) and references therein. An alternative is the book by Tanner (1996). However, when we turn to hypothesis testing, the approaches have little in common. Bayesians (and many like-minded thinkers) have a deep and abiding antipathy toward p-values, for a long list of reasons we shall not go into here; the interested reader may consult Berger (1985, Section 4.3.3), Kass and Raftery (1995, Section 8.2), or Carlin and Louis (2000, Section 2.3.3).

1.3.1 Point estimation

To keep things simple, suppose for the moment that θ is univariate. Given the posterior $p(\theta|\mathbf{y})$, a sensible Bayesian point estimate of θ would be some measure of centrality. Three familiar choices are the posterior mean,

$$\hat{\theta} = E(\theta|\mathbf{y}),$$

the posterior median,

$$\hat{\theta} \int_{-\infty}^{\hat{\theta}} p(\theta|\mathbf{y})d\theta = 0.5,$$

and the posterior mode,

$$\hat{\theta} p(\hat{\theta}|\mathbf{y}) = \sup_{\theta} p(\theta|\mathbf{y}).$$

Notice that the lattermost estimate is typically easiest to compute, since it does not require any integration: we can replace $p(\theta|\mathbf{y})$ by its unstandardized form, $f(\mathbf{y}|\theta)p(\theta)$, and get the same answer (since these two differ only by a multiplicative factor of $m(\mathbf{y})$, which does not depend on θ). Indeed, if the posterior exists under a flat prior $p(\theta) = 1$, then the posterior mode is nothing but the maximum likelihood estimate (MLE). Note that for symmetric unimodal posteriors (e.g. a normal distribution), the posterior mean, median, and mode will all be equal. However, for multimodal or otherwise non-normal posteriors, the mode will often be the poorest choice of centrality measure (consider, for example, the case of a steadily decreasing, one-tailed posterior; the mode will be the smallest value in the support of the distribution—hardly central!). By contrast, the posterior mean will sometimes be overly influenced by heavy tails (just as the sample mean $\bar{y}$ is not robust against outlying observations). As a result, the posterior median will often be the best and safest point estimate. It is also the most difficult to compute (since it requires both an integration and a root-finder), but this difficulty is mitigated for posterior estimates computed via MCMC; see Section 1.5.

1.3.2 Interval estimation

The posterior allows us to make direct probability statements not only regarding the median, but for any quantile. For example, suppose we can find the $\alpha/2$- and $(1-\alpha/2)$-quantiles of $p(\theta|\mathbf{y})$, that is, the points θ_{L} and θ_{U} such that

$$\int_{-\infty}^{\theta_{\mathrm{L}}} p(\theta|\mathbf{y})d\theta = \alpha/2$$

and

$$\int_{\theta_{\mathrm{U}}}^{\infty} p(\theta|\mathbf{y})d\theta = 1-\alpha/2.$$

Then clearly $P(\theta_L < \theta < \theta_U|\mathbf{y}) = 1-\alpha$; our confidence that θ lies in (θ_L, θ_U) is $100 \times (1-\alpha)\%$. Thus this interval is a $100 \times (1-\alpha)\%$ *credible set* (or simply *Bayesian confidence interval*) for θ. This interval is relatively easy to compute, and enjoys a direct interpretation ("the probability that θ lies in (θ_L, θ_U) is $(1-\alpha)$") that the usual frequentist interval does not. The interval just described is often called the *equal tail* credible set, for the obvious reason that is obtained by chopping an equal amount of support $(\alpha/2)$ off the top and bottom of $p(\theta|\mathbf{y})$. Note that for symmetric unimodal posteriors, this equal tail interval will be symmetric about this mode (which we recall equals the mean and median in this case). It will also be optimal in the sense that it will have shortest length among sets C satisfying

$$1-\alpha \leq P(C|\mathbf{y}) = \int_C p(\theta|\mathbf{y})d\theta. \quad (1.3)$$

Note that any such set C could be thought of as a $100 \times (1-\alpha)\%$ credible set for θ. For posteriors that are not symmetric and unimodal, a better (shorter) credible set can be obtained by taking only those values of θ having posterior density greater than some cutoff $k(\alpha)$, where this cutoff is chosen to be as large as possible while C still satisfies equation (1.3). This *highest posterior density* (HPD) confidence set will always be of minimal length, but will typically be much more difficult to compute. The equal tail interval emerges as HPD in the symmetric unimodal case since there too it captures the "most likely" values of θ. The equal tail interval estimate is most widely used with hierarchical models since it is easily obtained from the posterior samples that are the output of simulation-based model fitting approaches. Fortunately, many of the posteriors we will be interested in will be (at least approximately) symmetric unimodal, so the equal tail interval will often suffice.

1.3.3 Hypothesis testing and model choice

While Bayesian estimation is quite straightforward given the posterior distribution, or an estimate thereof, hypothesis testing is less straightforward, for two reasons. First, there is less agreement among Bayesians as to the proper approach to the problem. For years, posterior probabilities and Bayes factors were considered the only appropriate method. But these methods are only suitable with fully proper priors, and for relatively low-dimensional models. With the recent proliferation of very complex models with at least partly improper priors, other methods have come to the fore. Second, solutions to hypothesis testing questions often involve not just the posterior $p(\boldsymbol{\theta}|\mathbf{y})$, but also the *marginal* distribution, $m(\mathbf{y})$. Unlike the case of posterior and the predictive distributions, samples from the marginal distribution do not naturally emerge from most MCMC algorithms. Thus, the sampler must often be "tricked" into producing the necessary samples. Recently, an approximate yet very easy-to-use model choice tool known as the Deviance Information Criterion (DIC) has gained popularity, as well as implementation in the `WinBUGS` software package. We will limit our attention in this subsection to Bayes factors, the DIC, and a related posterior predictive criterion due to Gelfand and Ghosh (1998). The reader is referred to Carlin and Louis (2000, Sections 2.3.3, 6.3, 6.4, and 6.5) for further techniques and information. We would also note that formal model choice reduces a model to a single number for comparison with numbers associated with other models. In practice, more informal comparison through displays of say, prediction or estimation performance may be more satisfying.

1.3.3.1 Bayes factors

We begin by setting up the hypothesis testing problem as a model choice problem, replacing the customary two hypotheses H_0 and H_A by two candidate parametric models M_1 and M_2 having respective parameter vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Under prior densities $\pi_i(\boldsymbol{\theta}_i)$, $i = 1, 2$, the marginal distributions of $\mathbf{Y}$ are found by integrating out the parameters,

$$p(\mathbf{y}|M_i) = \int f(\mathbf{y}|\boldsymbol{\theta}_i, M_i)\pi_i(\boldsymbol{\theta}_i)d\boldsymbol{\theta}_i, \quad i = 1, 2. \quad (1.4)$$

Bayes' Theorem (1.1) may then be applied to obtain the posterior probabilities $P(M_1|\mathbf{y})$ and $P(M_2|\mathbf{y}) = 1 - P(M_1|\mathbf{y})$ for the two models. The quantity commonly used to summarize these results is the *Bayes factor*, BF, which is the ratio of the posterior odds of M_1 to the prior odds of M_1, given by Bayes'

Theorem as

$$\text{BF} = \frac{P(M_1|\mathbf{y})/P(M_2|\mathbf{y})}{P(M_1)/P(M_2)} \tag{1.5}$$

$$= \left[\frac{(p(\mathbf{y}|M_1)P(M_1))/p(\mathbf{y})}{(p(\mathbf{y}|M_2)P(M_2))/p(\mathbf{y})}\right]\left[\frac{P(M_1)}{P(M_2)}\right]^{-1}$$

$$= \frac{p(\mathbf{y}|M_1)}{p(\mathbf{y}|M_2)}, \tag{1.6}$$

the ratio of the observed marginal densities for the two models. Assuming the two models are a priori equally probable (i.e. $P(M_1) = P(M_2) = 0.5$), we have that $\text{BF} = P(M_1|\mathbf{y})/P(M_2|\mathbf{y})$, the posterior odds of M_1.

Consider the case where both models are *simple*, that is, the priors put mass one on say $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_{10}$ and $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_{20}$, respectively. Then from (1.4) and (1.6) we have

$$\text{BF} = \frac{f(\mathbf{y}|\boldsymbol{\theta}_{10})}{f(\mathbf{y}|\boldsymbol{\theta}_{20})},$$

which is nothing but the likelihood ratio between the two models. Hence, in the simple-versus-simple setting, the Bayes factor is precisely the odds in favor of M_1 over M_2 *given solely by the data*.

In the case of nested nonhierarchical models, say M_1 of dimension p_1 contained in M_2 of dimension p_2, the *Bayesian Information Criterion* (BIC) (also known as the *Schwarz Criterion*), is given by

$$\Delta\text{BIC} = W - (p_2 - p_1)\log\, n, \tag{1.7}$$

where p_i is the number of parameters in model $M_i, i = 1, 2$, and

$$W = -2\log\left[\frac{\sup_{M_1} f(\mathbf{y}|\boldsymbol{\theta})}{\sup_{M_2} f(\mathbf{y}|\boldsymbol{\theta})}\right],$$

the usual likelihood ratio test statistic. Schwarz (1978) showed that, in this case, for large sample sizes n, BIC approximates $-2\log \text{BF}$. An alternative to BIC is the *Akaike Information Criterion* (AIC)*j* which alters (1.7) to

$$\Delta\text{AIC} = W - 2(p_2 - p_1). \tag{1.8}$$

Both AIC and BIC are *penalized likelihood ratio* model choice criteria, since both have second terms that act as a penalty, correcting for differences in size between the models. Crucially, the BIC penalty depends upon sample size while the AIC penalty does not. The implication is that for the former, the penalty tends to ∞ as $n \to \infty$ while for the latter it is constant, regardless of sample size. The upshot is that, under mild conditions, BIC is consistent, that is, the probability that model M_1 is selected when it is, in fact, true tends to ∞ as $n \to \infty$; AIC is not consistent. Expressed in a different way, AIC tends to favor more complex models than does BIC.

Another limitation in using Bayes factors or their approximations is that they are not appropriate under noninformative priors. To see this, note that if $\pi_i(\boldsymbol{\theta}_i)$ is improper, then $p(\mathbf{y}|M_i) = \int f(\mathbf{y}|\boldsymbol{\theta}_i, M_i)\pi_i(\boldsymbol{\theta}_i)d\boldsymbol{\theta}_i$ necessarily is as well, and so BF as given in (1.6) is not well defined. Also, for the hierarchical models we are interested in, we have no simple methods or approximations to compute Bayes factors. Instead, we offer two alternatives which are applicable to general hierarchical models, are easily computed using output from posterior simulation, and have achieved some popularity. Both can be criticized but, in reality, there will never be universal agreement on a model selection criterion since different researchers have different utilities for models.

1.3.3.2 The DIC criterion

Spiegelhalter et al. (2002) propose a generalization of the AIC, (Akaike 1973) based on the posterior distribution of the *deviance* statistic,

$$D(\boldsymbol{\theta}) = -2\log f(\mathbf{y}|\boldsymbol{\theta}) + 2\log\, h(\mathbf{y}), \tag{1.9}$$

where $f(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood function and $h(\mathbf{y})$ is some standardizing function of the data alone. These authors suggest summarizing the *fit* of a model by the posterior expectation of the deviance, $\overline{D} = E_{\theta|y}[D]$, and the *complexity* of a model by the effective number of parameters p_{D} (which may well be less than the total number of model parameters, due to the borrowing of strength across random effects). In the case of Gaussian models, one can show that a reasonable definition of p_{D} is the expected deviance minus the deviance evaluated at the posterior expectations,

$$p_{\text{D}} = E_{\theta|y}[D] - D(E_{\theta|y}[\boldsymbol{\theta}]) = \overline{D} - D(\bar{\boldsymbol{\theta}}). \tag{1.10}$$

The *Deviance Information Criterion* (DIC) is then defined as

$$\text{DIC} = \overline{D} + p_{\text{D}} = 2\overline{D} - D(\bar{\boldsymbol{\theta}}), \tag{1.11}$$

with smaller values of DIC indicating a better-fitting model. Both building blocks of DIC and p_{D}, $E_{\theta|y}[D]$ and $D(E_{\theta|y}[\boldsymbol{\theta}])$, are easily estimated via MCMC methods (see below), enhancing the approach's appeal. Indeed, DIC may be computed automatically for any model in the `WinBUGS` software (see Section 1.6). While the p_{D} portion of this expression does have meaning in its own right as an effective model size, DIC itself does not, since it has no absolute scale (due to the arbitrariness of the scaling constant $h(\mathbf{y})$, which is often simply set equal to zero). Thus only *differences* in DIC across models are meaningful. In this regard, when DIC is used to compare nested models in standard exponential family settings, the likelihood $L(\boldsymbol{\theta}; \mathbf{y})$ is often used in place of the normalized form $f(\mathbf{y}|\boldsymbol{\theta})$ in (1.9). This is appropriate since, in this case, for a fixed $\mathbf{y}$ the former is a constant times the latter and this constant does not change across models. Hence, on the log scale it contributes equally to the DIC scores of each (and thus has no impact on model selection). However, in settings where we require comparisons across different likelihood forms, that is, the competing models have data generating mechanisms that come from different distributional families, generally one must be careful to use the properly scaled joint density $f(\mathbf{y}|\boldsymbol{\theta})$. Indeed, we are most comfortable recommending the use of DIC for comparison of models employing the same first stage likelihood.

Identification of what constitutes a *significant* difference is also a bit awkward; delta method approximations to Var(DIC) have to date met with little success (Zhu and Carlin 2000). In practice one typically adopts the informal approach of simply recomputing DIC a few times using different random number seeds, to get a rough idea of the variability in the estimates. With a large number of independent DIC replicates $\{\text{DIC}_l, l = 1, \ldots, N\}$, one could of course estimate Var(DIC) by its sample variance,

$$\widehat{\text{Var}}(\text{DIC}) = \frac{1}{N-1}\sum_{l=1}^{N}(\text{DIC}_l - \overline{\text{DIC}})^2.$$

But in any case, DIC is not intended for formal identification of the "correct" model, but rather merely as a method of comparing a collection of alternative formulations (all of which will likely be incorrect). This informal outlook (and DIC's approximate nature in markedly nonnormal models) suggests informal measures of its variability will often be sufficient. The p_{D} statistic is also helpful in its own right, since how close it is to the actual parameter count provides information about how many parameters are actually "needed" to adequately explain the data. For instance, a relatively low p_{D} may indicate collinear fixed effects or lots of borrowing of strength across random effects. DIC is remarkably general, and trivially computed as part of an MCMC run without any need for extra sampling, reprogramming, or complicated loss function determination. Moreover, experience with DIC to date suggests it works remarkably well, despite the fact that no formal justification for it is yet available outside of posteriors that can be well approximated by a Gaussian distribution (a condition that typically occurs asymptotically, but perhaps not without a moderate to large sample size for many models). Still, DIC is by no means universally accepted by Bayesians as a suitable all-purpose model choice tool, as the discussion to Spiegelhalter et al. (2002) directly indicates.

Model comparison using DIC is not invariant to parametrization, so (as with prior elicitation) the most sensible parametrization must be carefully chosen beforehand. Unknown scale parameters and other innocuous restructuring of the model can also lead to subtle changes in the computed DIC value.

Finally, DIC will obviously depend on what part of the model specification is considered to be part of the likelihood, and what is not. Spiegelhalter et al. (2002) refer to this as the *focus* issue, that is, determining which parameters are of primary interest, and which should "count" in p_{D}. For instance, in a hierarchical model with data distribution $f(\mathbf{y}|\boldsymbol{\theta})$, prior $p(\boldsymbol{\theta}|\eta)$ and hyperprior $p(\eta)$, one might choose as the likelihood either the obvious conditional expression $f(\mathbf{y}|\boldsymbol{\theta})$, or the *marginal* expression,

$$p(\mathbf{y}|\eta) = \int f(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\eta)d\boldsymbol{\theta}. \tag{1.12}$$

We refer to the former case as "focused on $\boldsymbol{\theta}$," and the latter case as "focused on η." Spiegelhalter et al. (2002) defend the dependence of p_{D} and DIC on

the choice of focus as perfectly natural, since while the two foci give rise to the same marginal density $m(y)$, the integration in (1.12) clearly suggests a different model complexity than the unintegrated version (having been integrated out, the θ parameters no longer "count" in the total). They thus argue that it is up to the user to think carefully about which parameters ought to be in focus before using DIC. Perhaps the one difficulty with this advice is that, in cases where the integration in (1.12) is not possible in closed form, the unintegrated version is really the only feasible choice. Indeed, the DIC tool in `WinBUGS` always focuses on the lowest level parameters in a model (in order to sidestep the integration issue), even when the user intends otherwise.

1.3.3.3 Posterior predictive loss criteria

An alternative to DIC that is also easily implemented using output from posterior simulation is the *posterior predictive loss* (performance) approach of Gelfand and Ghosh (1998). Using prediction with regard to replicates of the observed data, $Y_{\ell,\text{rep}}, \ell = 1, \ldots, n$, the selected models are those that perform well under a so-called *balanced* loss function. Roughly speaking, this loss function penalizes actions both for departure from the corresponding observed value ("fit") as well as for departure from what we expect the replicate to be ("smoothness"). The loss puts weights $k > 0$ and 1 on these two components, respectively, to allow for relative weighting of regret (or loss) for the two types of departure. We avoid details here, but note that for squared error loss, the resulting criterion becomes

$$D_k = \frac{k}{k+1} G + P, \tag{1.13}$$

where

$$G = \sum_{\ell=1}^{n} (\mu_\ell - y_{\ell,\text{obs}})^2$$

and

$$P = \sum_{\ell=1}^{n} \sigma_\ell^2.$$

In (1.13), $\mu_\ell = E(Y_{\ell,\text{rep}}|\mathbf{y})$ and $\sigma_\ell^2 = \text{Var}(Y_{\ell,\text{rep}}|\mathbf{y})$, that is, the mean and variance of the predictive distribution of $Y_{\ell,\text{rep}}$ given the observed data $\mathbf{y}$. The components of D_k have natural interpretations. G is a goodness-of-fit term, while P is a penalty term. To clarify, we are seeking to penalize complexity and reward parsimony, just as DIC and other penalized likelihood criteria do. For a poor model we expect large predictive variance and poor fit. As the model improves, we expect to do better on both terms. But as we start to overfit, we will continue to do better with regard to goodness of fit, but also begin to inflate the variance (as we introduce multicollinearity). Eventually the resulting increased predictive variance penalty will exceed the gains in goodness-of-fit. So as with DIC, as we sort through a collection of models, the one with the smallest D_k is preferred. When $k = \infty$ (so that $D_k = D_\infty = G + P$), we will sometimes write D_∞ simply as D for brevity.

Two remarks are appropriate. First, we may report the first and second terms (excluding $k/(k+1)$) on the right side of (1.13), rather than reducing to the single number D_k. Second, in practice, ordering of models is typically insensitive to the particular choice of k. The quantities μ_ℓ and σ_ℓ^2 can be readily computed from posterior samples. If under model m we have parameters $\boldsymbol{\theta}^{(m)}$, then

$$p(y_{\ell,\text{rep}}|\mathbf{y}) = \int p(y_{\ell,\text{rep}}|\boldsymbol{\theta}^{(m)}) p(\boldsymbol{\theta}^{(m)}|\mathbf{y}) d\boldsymbol{\theta}^{(m)}. \tag{1.14}$$

Hence each posterior realization (say, $\boldsymbol{\theta}^*$) can be used to draw a corresponding $y_{\ell,\text{rep}}$ from $p(y_{\ell,\text{rep}}|\boldsymbol{\theta}^{(m)} = \boldsymbol{\theta}^*)$. The resulting $y^*_{\ell,\text{rep}}$ has marginal distribution $p(y_{\ell,\text{rep}}|\mathbf{y})$. With samples from this distribution we can obtain μ_ℓ and σ_ℓ^2. That is, development of D_k requires an extra level of simulation but this can be done after the model has been fitted. More precisely, once we have the posterior samples, we can obtain draws of the set of $\{y_{l,\text{rep}}\}$ one for one with these samples. More general loss functions can be used, including the so-called deviance loss (based upon $p(y_\ell|\boldsymbol{\theta}^{(m)})$), again yielding two terms for D_k with corresponding interpretation and predictive calculation. This enables application to, say, binomial or Poisson likelihoods. We omit details here since in this book, only (1.13) is used for examples that employ this criterion rather than DIC. We do not recommend a choice between the posterior predictive approach of this subsection and the DIC of

AQ: Please check should l be ℓ in $\{y_{l,\text{rep}}\}$.

the previous subsection. Both involve summing a goodness-of-fit term and a complexity penalty. The fundamental difference is that the DIC works in the parameter space with the likelihood, while predictive loss works in predictive space with posterior predictive distributions. The DIC addresses comparative explanatory performance, while predictive loss addresses comparative predictive performance. So, if the objective is to use the model for explanation, we may prefer DIC; if instead the objective is prediction, we may prefer D_k.

1.4 Hierarchical models

A key goal of modeling for many complex biological process is the development of a multilevel stochastic specification that is built from local, simple relationships but, in total, captures the important components in explaining the behavior of the process. This is the essence of the hierarchical modeling that will be at the heart of the various presentations in the ensuing chapters. Here we attempt a brief introduction to such modeling.

It can be pragmatic to view modeling problems in terms of three entities, all of which have stochastic elements. First is the *data* which is presumed to be drawn from some facet(s) of the underlying process. Second is the *process* specification itself which involves unknowns that will be estimated as parameters. Third, we have *parameters* that are not only "uncertain" but will be expected to vary depending upon how and where the data were obtained. With this three-part structure in mind, we are prepared to extend the earlier version of the Bayesian model to more levels in a general and flexible way. Because stochasticity is relevant for each, we think in terms of a joint distribution

$$\begin{aligned} &f(\text{data,process,parameters}) \\ &\quad \propto f(\text{data}|\text{process, parameters}) \\ &\qquad \times f(\text{process}|\text{parameters}) \\ &\qquad \times f(\text{parameters}). \end{aligned}$$

The joint distribution on the left side is provided in terms of three pieces on the right side. These pieces may be easier to consider individually rather than thinking about the entire joint distribution. Moreover, as the chapters that follow will reveal, each of these pieces can be quite complex. For instance, the relationship between data and process might depend on many things. It might be different for different types of data. There may be spatial or temporal aspects that suggest the modeling might depend upon where and when the process occurred. The good news is that we can use appropriate conditioning to capture these aspects in straightforward ways.

Advantages of this way of thinking about modeling include: (1) the ability to construct complex models from simple conditional relationships. We need not conceptualize an integrated specification for the problem, only the components which will be linked up through boxes, circles, and arrows (see below). (2) We can relax customary requirements for independent data. Conditional independence is enough. We typically introduce dependence at a second or third stage in the modeling which, marginally, introduces association in the data. We can accommodate different data types within the analysis as well as "data" that are output from, say, a computer model. (3) By attaching randomness to what we observe as well as to what we do not observe, we build a fully Bayesian specification. The inherent unification of Bayesian inference leads immediately to looking at the posterior distribution of everything that we did not observe given everything that we did. Though such a posterior will be high dimensional and analytically intractable, we can take advantage of the Bayesian computation tools described in Section 1.5 to fit these models and provide the desired inference.

In general, the complex process model can be represented in the form of an acyclic dependence graph. We briefly describe such models in the context of the discussion of the previous paragraphs. For a full, accessible development of graphical models the reader is referred to Whitaker (1991). A more technical development is provided in Cowell et al. (1999). The ensuing chapters will provide illustrative graphical models in the course of developing their particular applications. We offer an illustrative one below.

A graphical model includes arrows and nodes (and may be viewed as a more formal version of "box and arrows" models that are familiar to ecologists). The nodes denote the variables that comprise or are

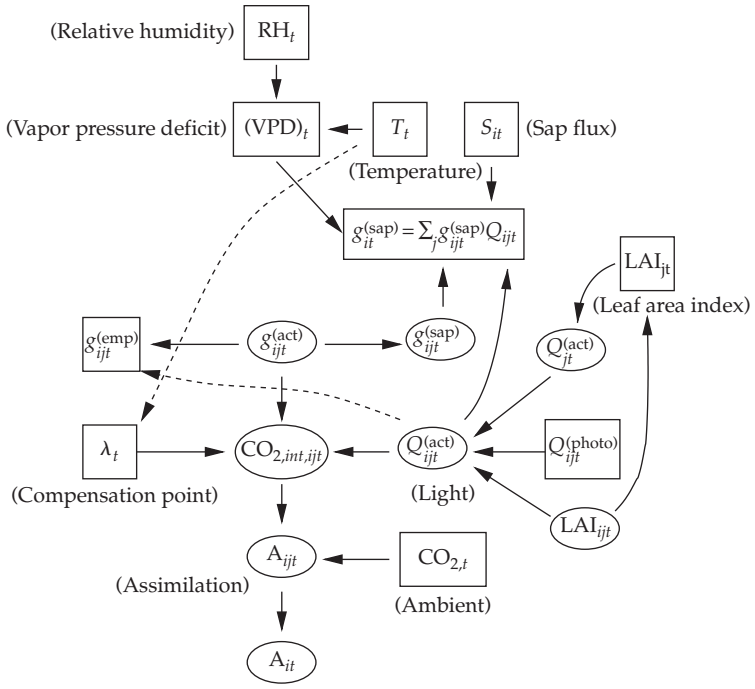

Figure 1.1. An illustrative graphical model.

used to explain the process. Some are observable, denoted by rectangles; some are conceptual or unobservable denoted by circles. The arrows to a node indicate which (input) nodes will be used to explain this (output) node. Some nodes arise deterministically, that is, an explicit functional relationship produces the output, given the inputs. Others are stochastic. We postulate a relationship between the inputs and the outputs but only expect that the output will be a random realization, given the inputs. So, there will be uncertainty associated with these nodes. In addition, we will not know the stochastic response model for the output, given the inputs explicitly. This local model will have parameters, for example, regression coefficients and a variance, additional model unknowns. The graph has source nodes (nodes with no inputs). These are not modelled. We proceed through the graph from inputs to outputs.

Figure 1.1 presents an illustrative graphical model which focuses on conductance (denoted by g) at the tree level in a stand of trees. Without detailing the experimental data or the model, the salient points are as follows. Conductance can be studied empirically to give estimates for particular trees (i), at particular heights (j), at particular times (t). However, these measurements are not well calibrated. Alternatively, sap flux measurements of conductance can be made at the tree level. These measurements are very accurate but must be disaggregated to particular heights. The basic objective of this graphical model is to assimilate these two sources of information to infer about the actual (unobservable) conductance which eventually leads to carbon assimilation. Relative humidity, vapor pressure deficit, temperature, leaf area index, and available light are available at different scales, as the subscripting indicates. So, we see through the boxes and circles what is observed and what we seek to infer about. Through the arrows we see which variables will be used to explain which responses. The explicit forms of the models are not presented; again, the goal is only to identify all the process variables of interest and to propose relationships among them.

The state of the process may evolve in time. Then, at any time t, there is a state for each node in the graph. To introduce temporal updating, some nodes at time t have to feed back into earlier nodes in the graph to update those nodes to a new state at time $t+1$. Source nodes update themselves. In this way the graph moves to a new state at time $t+1$.

The power of a graphical modeling approach for a complex system lies in its ability to provide a conceptual decomposition of the system into submodels, consistent with our foregoing discussion. Attention can be focused on trying to build suitable local models and then all of the pieces can be put together to create the system level model. Historically, researchers will likely have investigated the local components. Through this work, lab experiments, field studies, first principles, etc. we will have some insight into how to propose the local modeling. In fact, the local modeling can be tuned as needed and, if we ultimately have no information to guide us, we can introduce empirical or phenomenological specifications. The advantage to analysis of a system level model is that, by allowing the local submodels to interact with on another, we enhance our learning about all submodels. In fact, the overall model implies a complex dependence structure across the graph. Arrows can be removed to examine more parsimonious specifications. They can be added to see if fuller explanation is needed for some nodes.

From an inferential perspective, we seek to learn about the circled nodes given the boxed nodes. Because there will be parameters associated with many of the nodes, we really seek the conditional distribution, f(unobserved nodes, parameters| observed nodes). The directed graph provides the required joint distribution or likelihood. So we know the conditional distribution up to a normalizing constant, which is the joint distribution, inserting the values of the observed nodes. Such a simplified summary understates the computational challenge involved in fitting such a complex graphical model. We complete the model specification with weak prior specifications for unknown model parameters.

We then attempt fitting using Gibbs sampling/ Markov chain Monte Carlo algorithms (Section 1.5). In fact, the process of fitting the full model also benefits from local modeling. That is, local models may be fitted and the results will enable us to acquire some feel for the magnitudes of model parameters in an enlarged model. Typically, some approximation in implementing the overall fitting may be required as well, in order to enable realistic computing times.

A particularly attractive feature of the graphical model is its ability to propagate uncertainty across the model. The uncertainty associated with an input node augments the variability attached to the resulting output nodes. As this occurs across the graph and, perhaps, over time, we achieve an assessment of the full uncertainty at any node or set of nodes. We can also consider conditional uncertainty for a portion of the graph as a result of fixing the values of other nodes. Such uncertainty will be smaller than the full uncertainty but can be useful in studying the response of local processes to external inputs of interest.

1.5 Bayesian computation

Here, we provide a brief introduction to Bayesian computing, at the level of the presentation in say, Carlin and Louis (2000). The explosion in Bayesian activity and computing power of the last decade or so has caused a similar explosion in the number of books in this area. The earliest comprehensive treatment was by Tanner (1996), with books by Gilks et al. (1996), Gamerman (1997), and Chen et al. (2000) offering updated and expanded discussions that are primarily Bayesian in focus. Also significant are the computing books by Robert and Casella (1999) and Liu (2001), which, while not specifically Bayesian, still emphasize Markov chain Monte Carlo methods typically used in modern Bayesian analysis.

Without doubt, the most popular computing tools in Bayesian practice today are MCMC methods. This is due to their ability (in principle) to break the "curse of dimensionality," to enable inference from posterior distributions of very high dimension, essentially by reducing the problem to one of recursively treating a sequence of lower-dimensional (often one dimensional) problems. Like traditional Monte Carlo methods, MCMC methods work by producing not a closed form for the posterior in (1.1), but a *sample* of values $\{\boldsymbol{\theta}^{(g)}, g = 1, \ldots, G\}$ from this distribution. While this obviously does not carry as much

information as a closed form expression, a histogram or kernel density estimate based on such a sample is typically sufficient for reliable inference. Moreover such an estimate can be made arbitrarily accurate merely by increasing the Monte Carlo sample size G. (Note, importantly, that this has nothing to do with the sample size of the observed data.) However, unlike traditional Monte Carlo methods, MCMC algorithms produce *correlated* samples from this posterior, since they arise from recursive draws from a particular Markov chain, the stationary distribution of which is the same as the posterior.

The convergence of the Markov chain to the correct stationary distribution can be guaranteed for an enormously broad class of posteriors, explaining MCMC's popularity. But this convergence is also the source of most of the difficulty in actually implementing MCMC procedures, for two reasons. First, it forces us to make a decision about when it is safe to stop the sampling algorithm and summarize its output, an issue known as *convergence diagnosis*. Second, it clouds the determination of the quality of the estimates produced (since they are based not on i.i.d. draws from the posterior, but on correlated samples. This is sometimes called the *variance estimation* problem, since a common goal here is to estimate the Monte Carlo variances (equivalently standard errors) associated with our MCMC-based posterior estimates. In the remainder of this section, we introduce the two most popular notions in developing MCMC algorithms, the Gibbs sampler and the Metropolis–Hastings algorithm. We then return to the convergence diagnosis and variance estimation problems.

1.5.1 The Gibbs sampler

Suppose our model features k parameters, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)'$. To implement the Gibbs sampler, we must assume that samples can be generated from each of the *full* or *complete* conditional distributions $\{p(\theta_i|\boldsymbol{\theta}_{j\neq i}, \mathbf{y}), i = 1, \ldots, k\}$ in the model. These distributions are always known up to proportionality constant since they take the form of the likelihood x prior with everything fixed but θ_i. That is, we insert the current values of $\boldsymbol{\theta}_{j\neq i}$ and the observed $\mathbf{y}$. Samples might be available directly (say, if the full conditional is a familiar forms, like a normal or gamma) or indirectly (say, via a rejection sampling approach). In this latter case two popular alternatives are the adaptive rejection sampling (ARS) algorithm of Gilks and Wild (1992), and the Metropolis algorithm described in the next subsection. We note that, under compatibility conditions (which usually hold in practice), the collection of full conditional distributions uniquely determines the joint posterior distribution, $p(\boldsymbol{\theta}|\mathbf{y})$, and hence all marginal posterior distributions $p(\theta_i|\mathbf{y}), i = 1, \ldots, k$. Given an arbitrary set of starting values $\{\theta_2^{(0)}, \ldots, \theta_k^{(0)}\}$, the algorithm proceeds as follows:

Gibbs sampler: For $(t \in 1 : T)$, repeat:

Step 1: Draw $\theta_1^{(t)}$ from $p(\theta_1|\theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots, \theta_k^{(t-1)}, \mathbf{y})$

Step 2: Draw $\theta_2^{(t)}$ from $p(\theta_2|\theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_k^{(t-1)}, \mathbf{y})$

$\vdots$

Step k: Draw $\theta_k^{(t)}$ from $p(\theta_k|\theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_{k-1}^{(t)}, \mathbf{y})$

Under mild regularity conditions that are generally satisfied for most statistical models (see, for example, Roberts and Smith 1993), one can show that the k-tuple, $(\theta_1^{(t)}, \ldots, \theta_k^{(t)})$, obtained at iteration t converges in distribution to a draw from the true joint posterior distribution $p(\theta_1, \ldots, \theta_k|\mathbf{y})$. This means that for t sufficiently large (say, bigger than t_0), $\{\boldsymbol{\theta}^{(t)}, t = t_0 + 1, \ldots, T\}$ is a (correlated) sample from the true posterior, from which any posterior quantities of interest may be estimated. For example, a histogram of the $\{\theta_i^{(t)}, t = t_0 + 1, \ldots, T\}$ themselves provides a simulation-consistent estimator of the marginal posterior distribution for θ_i, $p(\theta_i|\mathbf{y})$. We might also use a sample mean to estimate the posterior mean, that is,

$$\widehat{E}(\theta_i|\mathbf{y}) = \frac{1}{T - t_0} \sum_{t=t_0+1}^{T} \theta_i^{(t)}. \tag{1.15}$$

The time from $t = 0$ to $t = t_0$ is commonly known as the *burn-in* period; popular methods for selection of an appropriate t_0 are discussed below.

In practice, we may actually run *m parallel* Gibbs sampling chains, instead of only 1, for some modest

m (say, $m = 5$). We will see below that such parallel chains may be useful in assessing sampler convergence, and anyway can be produced with no extra time on a multiprocessor computer. In this case, we would again discard all samples from the burn-in period, obtaining the posterior mean estimate,

$$\widehat{E}(\theta_i|\mathbf{y}) = \frac{1}{m(T-t_0)} \sum_{j=1}^{m} \sum_{t=t_0+1}^{T} \theta_{i,j}^{(t)}, \tag{1.16}$$

where now the second subscript on $\theta_{i,j}$ indicates chain number. Again we defer comment on how the issues how to choose t_0 and how to assess the quality of (1.16) and related estimators for the moment. As a historical footnote, we add that Geman and Geman (1984) apparently introduced the name "Gibbs sampler" because the distributions used in their context (image restoration, where the parameters were actually the colors of pixels on a screen) were Gibbs distributions. These were, in turn, named after J.W̃.G̃ibbs, a nineteenth-century American physicist and mathematician generally regarded as one of the founders of modern thermodynamics and statistical mechanics. While Gibbs distributions form an exponential family on potentials that includes most standard statistical models as special cases, most Bayesian applications do not require anywhere near this level of generality, typically dealing solely with standard statistical distributions (normal, gamma, etc.). Yet, despite a few attempts by some Bayesians to choose a more descriptive name (e.g. the "successive substitution sampling" (SSS) moniker due to Schervish and Carlin 1992), the Gibbs sampler name has stuck.

1.5.2 The Metropolis–Hastings algorithm

The Gibbs sampler is easy to understand and implement, but requires the ability to readily sample from each of the full conditional distributions, $p(\theta_i|\boldsymbol{\theta}_{j\neq i}, \mathbf{y})$. Unfortunately, when the prior distribution $p(\boldsymbol{\theta})$ and the likelihood $f(\mathbf{y}|\boldsymbol{\theta})$ are not a conjugate pair, one or more of these full conditionals may not be available in closed form. As noted above, even in this setting, $p(\theta_i|\boldsymbol{\theta}_{j\neq i}, \mathbf{y})$ *will* be available up to a proportionality constant, since it is proportional to the portion of $f(\mathbf{y}|\boldsymbol{\theta}) \times p(\boldsymbol{\theta})$ that involves θ_i. The *Metropolis algorithm* (or *Metropolis–Hastings algorithm*) is a rejection algorithm that attacks precisely this problem, since it requires only a function proportional to the distribution to be sampled, at the cost of requiring a rejection step from a particular *candidate* density. Like the Gibbs sampler, this algorithm was not developed by statistical data analysts for this purpose, but by statistical physicists working on the Manhattan Project in the 1940s seeking to understand the particle movement theory underlying the first atomic bomb (see, for example, the seminal paper in this area, Metropolis et al. 1953).

While, as mentioned above, our main interest in the algorithm is for generation from (typically univariate) full conditionals, for convenience, we describe it for the full multivariate θ vector. Thus, suppose for now that we wish to generate from a joint posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}) \propto h(\boldsymbol{\theta}) \equiv f(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. We begin by specifying a candidate density $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$ that is a valid density function for every possible value of the conditioning variable $\boldsymbol{\theta}^{(t-1)}$, and satisfies $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)$, that is, q is *symmetric* in its arguments. Most naturally, we would take q of the form $q(\boldsymbol{\theta}^* - \boldsymbol{\theta}^{(t-1)})$. Then, given a starting value $\boldsymbol{\theta}^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

Metropolis algorithm: For $(t \in 1 : T)$, repeat:

Step 1: Draw $\boldsymbol{\theta}^*$ from $q(\cdot|\boldsymbol{\theta}^{(t-1)})$
Step 2: Compute the ratio $r = h(\boldsymbol{\theta}^*)/h(\boldsymbol{\theta}^{(t-1)}) = \exp[\log h(\boldsymbol{\theta}^*) - \log h(\boldsymbol{\theta}^{(t-1)})]$
Step 3: If $r \geq 1$, set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^*$; If $r < 1$, set

$$\boldsymbol{\theta}^{(t)} = \begin{cases} \boldsymbol{\theta}^* & \text{with probability } r, \\ \boldsymbol{\theta}^{(t-1)} & \text{with probability } 1-r \end{cases}$$

Then under generally the same mild conditions as those supporting the Gibbs sampler, draws $\boldsymbol{\theta}^{(t)}$ converge in distribution to a draw from the true posterior density $p(\boldsymbol{\theta}|\mathbf{y})$. Note however that when the Metropolis algorithm (or the Metropolis–Hastings algorithm below) is used to update within a Gibbs sampler, it never samples from the full conditional distribution. Convergence using Metropolis steps, then, would be expected to be slower than that for a regular Gibbs sampler. Recall that the steps of the Gibbs sampler were fully determined by the statistical model under consideration (since full conditional distributions for well defined models

are unique). By contrast, the Metropolis algorithm affords substantial flexibility through the selection of the candidate density q. This flexibility can be a blessing and a curse: while theoretically we are free to pick almost anything, in practice only a "good" choice will result in sufficiently many candidate acceptances. The usual approach (after $\boldsymbol{\theta}$ has been transformed to have support $\Re^k$, if necessary) is to set

$$q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = N(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}, \widetilde{\Sigma}), \tag{1.17}$$

since this distribution obviously satisfies the symmetry property, and is "self correcting" (candidates are always centered around the current value of the chain). Specification of q then comes down to specification of $\widetilde{\Sigma}$. Here we might try to mimic the posterior variance by setting $\widetilde{\Sigma}$ equal to an empirical estimate of the true posterior variance, derived from a preliminary sampling run.

The form in (1.17) is referred to as a "random walk" proposal; we propose a random mean 0 increment to the current $\boldsymbol{\theta}^{(t-1)}$. The reader might well imagine an optimal choice of q would produce an empirical acceptance ratio of 1, the same as the Gibbs sampler (and with no apparent "waste" of candidates). However, the issue is rather more subtle than this: accepting all or nearly all of the candidates is often the result of an overly narrow candidate density. Such a density will "baby-step" around the parameter space, leading to high acceptance but also high autocorrelation in the sampled chain. An overly wide candidate density will also struggle, proposing leaps to places far from the bulk of the posterior's support, leading to high rejection and again, high autocorrelation. Thus the "folklore" here is to choose $\widetilde{\Sigma}$ so that roughly 50% of the candidates are accepted. Subsequent theoretical work (e.g. Gelman et al. 1996) indicates even lower acceptance rates (25–40%) are optimal. This result varies with the dimension and true posterior correlation structure of $\boldsymbol{\theta}$ but provides a useful benchmark when developing your own Metropolis algorithm code. As a result, choice of $\widetilde{\Sigma}$ is often done *adaptively*. For instance, in one dimension (setting $\widetilde{\Sigma} = \widetilde{\sigma}$, and thus avoiding the issue of correlations among the elements of $\boldsymbol{\theta}$), a common trick is to simply pick some initial value of $\widetilde{\sigma}$, and then keep track of the empirical proportion of candidates that are accepted. If this fraction is too high (75–100%), we simply increase $\widetilde{\sigma}$; if it is too low (0–20%), we decrease it. Since certain kinds of adaptation can actually disturb the chain's convergence to its stationary distribution, the simplest approach is to allow this adaptation only during the burn-in period, a practice sometimes referred to as *pilot adaptation*. This is in fact the approach currently used by `WinBUGS`, where the pilot period is fixed at 4000 iterations.

As mentioned above, in practice the Metropolis algorithm is often found as a substep in a larger Gibbs sampling algorithm, used to generate from awkward full conditionals. Such hybrid Gibbs–Metropolis applications were once known as "Metropolis within Gibbs" or "Metropolis substeps," and users would worry about how many such substeps should be used. Fortunately, it was soon realized that a single substep was sufficient to ensure convergence of the overall algorithm, and so this is now standard practice: when we encounter an awkward full conditional (say, for θ_i), we simply draw one Metropolis candidate, accept or reject it, and move on to θ_{i+1}. Further discussion of convergence properties and implementation of hybrid MCMC algorithms can be found in Tierney (1994) and Carlin and Louis (2000, Section 5.4.4).

We end this subsection with the important generalization of the Metropolis algorithm devised by Hastings (1970). In this variant we drop the requirement that q be symmetric in its arguments, which is often useful for bounded parameter spaces (say, $\theta > 0$) where Gaussian proposals as in (1.17) are not natural.

Metropolis–Hastings algorithm: In Step 2 of the Metropolis algorithm earlier, replace the acceptance ratio r by

$$\begin{aligned} r &= \frac{h(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}{h(\boldsymbol{\theta}^{(t-1)})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})} \\ &= \exp[\log h(\boldsymbol{\theta}^*) - \log h(\boldsymbol{\theta}^{(t-1)}) + \cdots - \cdots]. \end{aligned} \tag{1.18}$$

Then again under mild conditions, draws $\boldsymbol{\theta}^{(t)}$ converge in distribution to a draw from the true posterior density $p(\boldsymbol{\theta}|\mathbf{y})$ as $t \to \infty$. In practice we often set

$$q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)}) = q(\boldsymbol{\theta}^*),$$

that is, we use a proposal density that ignores the current value of the variable. This algorithm is sometimes referred to as a *Hastings independence chain*, so named because the proposals (though not the final $\boldsymbol{\theta}^{(t)}$ values) form an independent sequence. While easy to implement, this algorithm can be difficult to tune since it will converge slowly unless the chosen q is rather close to the true posterior. In fact, it is evident that movement of the chain depends on the ratio of h to q at the proposed $\boldsymbol{\theta}$ relative to the ratio at the current $\boldsymbol{\theta}$ so q plays the role of an importance sampling density. See, for example, the books by Robert and Casella (1999) and Liu (2001) in this regard.

1.5.3 Slice sampling

An alternative to the Metropolis–Hastings algorithm that is still quite general is *slice sampling* (Neal 2003). In this regard the general paper by Damien et al. (1999) and the spatial modeling oriented paper by Agarwal and Gelfand (2005) may be of interest. In its most basic form, suppose we seek to sample a univariate $\theta \sim f(\theta) \equiv h(\theta)/\int h(\theta)d\theta$, where $h(\theta)$ is known. Suppose we add a so-called *auxiliary variable* U such that $U|\theta \sim Unif(0, h(\theta))$. Then the joint distribution of θ and U is $p(\theta, u) \propto 1 \cdot I(U < h(\theta))$, where I denotes the indicator function. If we run a Gibbs sampler drawing from $U|\theta$ followed by $\theta|U$ at each iteration, we can obtain samples from $p(\theta, u)$, and hence from the marginal distribution of θ, $f(\theta)$. Sampling from $\theta|u$ requires a draw from a uniform distribution for θ over the set $S_U = \{\theta : U < h(\theta)\}$. Figure 1.2 provides an illustrative picture for a bimodal univariate density to reveal why this approach is referred to as slice sampling. U "slices" the nonnormalized density, and the resulting "footprint" on the axis provides S_U. If we can enclose S_U in an interval, we can draw θ uniformly on this interval and simply retain it only if $U < h(\theta)$ (i.e. if $\theta \in S_U$). If $\boldsymbol{\theta}$ is instead multivariate, S_U is more complicated and now we would need a bounding rectangle.

Note that if $h(\theta) = h_1(\theta)h_2(\theta)$ where, say, h_1 is a standard density that is easy to sample, while h_2 is nonstandard and difficult to sample, then we can introduce an auxiliary variable U such that $U|\theta \sim U(0, h_2(\theta))$. Now $p(\theta, u) = h_1(\theta)I(U < h_2(\theta))$. Again $U|\theta$ is routine to sample, while to sample $\theta|U$ we would now draw θ from $h_1(\theta)$ and retain it only if θ is such that $U < h_2(\theta)$. Slice sampling incurs problems similar to rejection sampling in that we may have to draw many θ's from h_1 before we are able to retain one. On the other hand, it has an advantage over the Metropolis–Hastings algorithm in that it always samples from the exact full conditional $p(\theta|u)$. As noted above, Metropolis–Hastings does not, and thus slice sampling would be expected to converge more rapidly. Nonetheless, overall comparison of computation time may make one method a winner for some cases, and the other a winner in other cases.

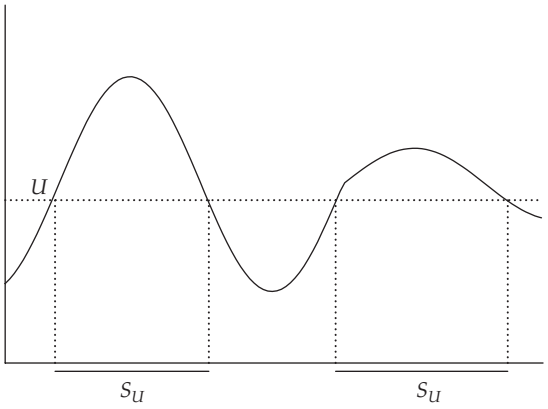

Figure 1.2. Illustrative slice sampling.

1.5.4 Convergence diagnosis

As mentioned above, the most problematic part of MCMC computation is deciding when it is safe to stop the algorithm and summarize the output. This means we must make a guess as to the iteration t_0 after which all output may be thought of as coming from the true stationary distribution of the Markov chain (i.e. the true posterior distribution). The most common approach here is to run a few (say, $m = 3$–5) *parallel* sampling chains, initialized at widely disparate starting locations that are overdispersed with respect to the true posterior. These chains are then plotted on a common set of axes, and the resulting *trace plots* are then viewed to see if there is an identifiable point t_0 after which all m chains seem to be "overlapping" (traversing the same part of $\boldsymbol{\theta}$-space).

Sadly, there are obvious problems with this approach. First, since the posterior is unknown at the outset, there is no reliable way to ensure that the m chains are "initially overdispersed," as required for a convincing diagnostic. We might use extreme quantiles of the prior $p(\boldsymbol{\theta})$ and rely on the fact that the support of the posterior is typically a subset of that of the prior, but this requires a proper prior and in any event is perhaps doubtful in high-dimensional or otherwise difficult problems. Second, it is hard to see how to automate such a diagnosis procedure, since it requires a subjective judgment call by a human viewer. A great many papers have been written on various convergence diagnostic statistics that summarize MCMC output from one or many chains that may be useful when associated with various stopping rules; see Cowles and Carlin (1996) and Mengersen et al. (1999) for reviews of many such diagnostics.

One of the most popular diagnostics is that of Gelman and Rubin (1992). Here, we run a small number (m) of parallel chains with different starting points that are "initially overdispersed" with respect to the true posterior. (Of course, since we do not know the true posterior before beginning there is technically no way to ensure this; still, the rough location of the bulk of the posterior may be discernible from known ranges, the support of the (proper) prior, or perhaps a preliminary posterior mode-finding algorithm.) Running the m chains for $2N$ iterations each, we then try to see whether the variation within the chains for a given parameter of interest λ approximately equals the total variation across the chains during the latter N iterations. Specifically, we monitor convergence by the estimated *scale reduction factor*,

$$\sqrt{\hat{R}} = \sqrt{\left(\frac{N-1}{N} + \frac{m+1}{mN}\,\frac{B}{W}\right)\frac{df}{df-2}}, \qquad (1.19)$$

where B/N is the variance between the means from the m parallel chains, W is the average of the m within-chain variances, and df is the degrees of freedom of an approximating t density to the posterior distribution. Equation (1.19) is the factor by which the scale parameter of the t density might shrink if sampling were continued indefinitely; the authors show it must approach 1 as $N \to \infty$.

The approach is fairly intuitive and is applicable to output from any MCMC algorithm. However, it focuses only on detecting bias in the MCMC estimator; no information about the *accuracy* of the resulting posterior estimate is produced. It is also an inherently univariate quantity, meaning it must be applied to each parameter (or parametric function) of interest in turn, although Brooks and Gelman (1998) extend the Gelman and Rubin approach in three important ways, one of which is a multivariate generalization for simultaneous convergence diagnosis of every parameter in a model. While the Gelman–Rubin–Brooks and other formal diagnostic approaches remain popular, in practice very simple checks often work just as well and may even be more robust against "pathologies" (e.g. multiple modes) in the posterior surface that may easily fool some diagnostics. For instance, sample autocorrelations in any of the observed chains can inform about whether slow traversing of the posterior surface is likely to impede convergence. Sample cross-correlations (i.e. correlations between two different parameters in the model) may identify ridges in the surface (say, due to collinearity between two predictors) that will again slow convergence; such parameters may need to be updated in multivariate blocks, or one of the parameters dropped from the model altogether. Combined with a visual inspection of a few sample trace plots, the user can at least get a good feel of whether posterior estimates produced by the sampler are likely to be reliable.

1.5.5 Variance estimation

An obvious criticism of Monte Carlo methods generally is that no two analysts will obtain the identical inference since they will not generate identical posterior samples. This makes assessment of the variance of these estimators crucial. Combined with a central limit theorem, the result would be an ability to test whether two Monte Carlo estimates were significantly different. For example, suppose we have a single chain of N post-burn-in samples of a parameter of interest λ, so that our basic posterior mean estimator (1.15) becomes $\hat{E}(\lambda|\mathbf{y}) = \hat{\lambda}_N = (1/N)\sum_{t=1}^{N} \lambda^{(t)}$. Assuming the samples comprising this estimator are independent, a variance estimate

for it would be given by

$$\widehat{\text{Var}}_{\text{iid}}(\hat{\lambda}_N) = s_\lambda^2/N = \frac{1}{N(N-1)}\sum_{t=1}^{N}(\lambda^{(t)} - \hat{\lambda}_N)^2, \tag{1.20}$$

that is the sample variance, $s_\lambda^2 = (1/N-1) \times \sum_{t=1}^{N}(\lambda^{(t)} - \hat{\lambda}_N)^2$, divided by N. But while this estimate is easy to compute, it would very likely be an *underestimate* due to positive autocorrelation in the MCMC samples. One can resort to *thinning*, which is simply retaining only every kth sampled value, where k is the approximate lag at which the autocorrelations in the chain become insignificant. However, MacEachern and Berliner (1994) show that such thinning from a stationary Markov chain always increases the variance of sample mean estimators, and is thus suboptimal. This is intuitive; it is never a good idea to throw away information (in this case, $(k-1)/k$ of our MCMC samples) just to achieve approximate independence among those that remain. A better alternative is to use all the samples, but in a more sophisticated way. One such alternative uses the notion of *effective sample size*, or ESS (Kass et al. 1998, p. 99). ESS is defined as

$$\text{ESS} = N/\kappa(\lambda),$$

where $\kappa(\lambda)$ is the *autocorrelation time* for λ, given by

$$\kappa(\lambda) = 1 + 2\sum_{k=1}^{\infty}\rho_k(\lambda), \tag{1.21}$$

where $\rho_k(\lambda)$ is the autocorrelation at lag k for the parameter of interest λ. We may estimate $\kappa(\lambda)$ using sample autocorrelations estimated from the MCMC chain. The variance estimate for $\hat{\lambda}_N$ is then

$$\begin{aligned}\widehat{\text{Var}}_{\text{ESS}}(\hat{\lambda}_N) &= s_\lambda^2/\text{ESS}(\lambda) \\ &= \frac{\kappa(\lambda)}{N(N-1)}\sum_{t=1}^{N}(\lambda^{(t)} - \hat{\lambda}_N)^2.\end{aligned}$$

AQ: Please clarify sentence 'Note that unless the $\lambda^{(t)}$ ...'.

Note that unless the $\lambda^{(t)}$ are uncorrelated, $\kappa(\lambda) > 1$ and $\text{ESS}(\lambda) < N$, so that $\widehat{\text{Var}}_{\text{ESS}}(\hat{\lambda}_N) > \widehat{\text{Var}}_{\text{iid}}(\hat{\lambda}_N)$, in concert with intuition. That is, since we have fewer than N effective samples, we expect some inflation in the variance of our estimate. In practice, the autocorrelation time $\kappa(\lambda)$ in (1.21) is often estimated simply by cutting off the summation when the magnitude of the terms first drops below some "small" value (say, 0.1). This procedure is simple but may lead to a biased estimate of $\kappa(\lambda)$. Gilks et al. (1996, pp. 50–51) recommend an *initial convex sequence estimator* mentioned by Geyer (1992) which, while while still output-dependent and slightly more complicated, actually yields a consistent (asymptotically unbiased) estimate here.

A final and somewhat simpler (though also more naive) method of estimating $\text{Var}(\hat{\lambda}_N)$ is through *batching*. Here we divide our single long run of length N into m successive batches of length k (i.e. $N = mk$), with batch means $B_1, \ldots, B_m$. Clearly $\hat{\lambda}_N = \bar{B} = (1/m)\sum_{i=1}^{m} B_i$. We then have the variance estimate

$$\widehat{\text{Var}}_{\text{batch}}(\hat{\lambda}_N) = \frac{1}{m(m-1)}\sum_{i=1}^{m}(B_i - \hat{\lambda}_N)^2, \tag{1.22}$$

provided that k is large enough so that the correlation between batches is negligible, and m is large enough to reliably estimate $\text{Var}(B_i)$. It is important to verify that the batch means are indeed roughly independent, say, by checking whether the lag 1 autocorrelation of the B_i is less than 0.1. If this is not the case, we must increase k (hence N, unless the current m is already quite large), and repeat the procedure. Regardless of which of the above estimates $\hat{V}$ is used to approximate $\text{Var}(\hat{\lambda}_N)$, a 95% confidence interval for $E(\lambda|\mathbf{y})$ is then given by

$$\hat{\lambda}_N \pm z_{0.025}\sqrt{\hat{V}},$$

where $z_{0.025} = 1.96$, the upper 0.025 point of a standard normal distribution. If the batching method is used with fewer than 30 batches, it is a good idea to replace $z_{0.025}$ by $t_{m-1,0.025}$, the upper 0.025 point of a t distribution with $m-1$ degrees of freedom. `WinBUGS` offers both naive (1.20) and batched (1.22) variance estimates.

1.6 Implementation via `WinBUGS`

In this subsection we provide an introduction to Bayesian data analysis in `WinBUGS`, the most general and well-developed Bayesian software package available to date. `WinBUGS` is the `Windows` successor to `BUGS`, a UNIX package whose name originally arose as a humorous acronym for `Bayesian inference Using`

Gibbs Sampling. The package is freely available from the website http://www.mrc-bsu.cam.ac.uk/ bugs/welcome.shtml. The software comes with a user manual, as well as two examples manuals that are enormously helpful for learning the language and various strategies for Bayesian data analysis.

WinBUGS has an interactive environment that enables the user to specify models hierarchically as well as perform Gibbs sampling to generate posterior samples. Convergence diagnostics, model checks and comparisons, and other helpful plots and displays are also available. We will now look at three WinBUGS code for a few illustrative problems.

1.6.1 Simple linear regression

We begin by considering the line example, which is used as the first illustration in the WinBUGS manual itself. Consider a set of five artificial data pairs (x_i, y_i): (1, 1), (2, 3), (3, 3), (4, 3), (5, 5). We wish to fit a simple linear regression of Y on X using the notation,

$$Y_i \sim N(\mu_i, \sigma^2), \quad \text{where } \mu_i = \alpha + \beta x_i.$$

As the WinBUGS code in Figure 1.3 illustrates, the language allows a concise expression of the model, where dnorm(a,b) denotes a normal distribution with mean *a* and *precision* (reciprocal of the variance) *b*, and dgamma(c,d) denotes a gamma distribution with mean c/d and variance c/d^2. The data means mu[i] are specified using a *logical* link (denoted by <-), instead of a *stochastic* one (denoted by ∼).

```
model
   {
   for(i in 1:N){
     Y[i] ~ dnorm(mu[i], tau)
     mu[i] <- alpha + beta * x[i]
     }
   sigma <- 1/sqrt(tau)
   alpha ~ dnorm(0, 1.0E-6)
   beta ~ dnorm(0, 1.0E-6)
   tau ~ dgamma(1.0E-3, 1.0E-3)
   }
```

Figure 1.3. WinBUGS code for the line example.

The second logical expression allows the standard deviation σ to be estimated.

The parameters in the Gibbs sampling order here will be α, β, and $\tau = (1/\sigma^2)$. All parameters are given proper but minimally informative prior distributions; namely, either normals with very small precisions (10^{-6}) or a gamma prior with both parameters equal to $\epsilon = 10^{-3}$ (so that the prior has mean 1 but variance 10^3).

We next need to load in the data. The data can be represented using S-plus or R object notation as list(x = c(1, 2, 3, 4, 5), Y = c(1, 3, 3, 3, 5), N = 5), or as a combination of an S-plus object and a rectangular array with labels at the head of the columns, like so:

list(N=5)

x[ ]	Y[ ]
1	1
2	3
3	3
4	3
5	5

Implementation of this code in WinBUGS is most easily accomplished by pointing and clicking through the menu on the Model/Specification, Inference/Samples, and Inference/Update tools; the reader may refer to www.statslab.cam.ac.uk/∼krice/winbugsthemovie.html for an easy-to-follow Flash introduction to these steps. WinBUGS may also be called by R; see the functions written by Andrew Gelman for this purpose at www.stat.columbia.edu/∼gelman/bugsR/, or the new BRugs package described at http:// mathstat.helsinki.fi/openbugs/ or the "Minnesota version" at http://www.biostat.umn.edu/∼brad/software/BRugs/BRugs_install.html.

1.6.2 Hierarchical Poisson failure rates

Here we consider a hierarchical model for failure rates arising from discrete failure counts Y_i arising during an elapsed time of t_i for similar but not identical systems $i = 1, \ldots, k$. The hierarchical

Table 1.1 Pump failure data (Gaver and O'Muircheartaigh 1987, *Technometrics*)

i	Y_i	t_i	r_i
1	5	94.320	0.053
2	1	15.720	0.064
3	5	62.880	0.080
4	14	125.760	0.111
5	3	5.240	0.573
6	19	31.440	0.604
7	1	1.048	0.954
8	1	1.048	0.954
9	4	2.096	1.910
10	22	10.480	2.099

model we adopt is

$$Y_i|\theta_i \overset{\text{ind}}{\sim} \text{Poisson}(\theta_i t_i),$$
$$\theta_i|\alpha,\beta \overset{\text{ind}}{\sim} \text{Gamma}(\alpha,\beta),$$
$$\alpha \sim \text{Exp}(\mu), \quad \text{and} \quad \beta \sim \text{Gamma}(c,d),$$

where μ, c, d, and the t_i are known, and Exp denotes the exponential distribution with mean μ. Note the gamma offers a conjugate hyperprior for β, but the exponential is not conjugate for α (indeed there is no conjugate form available here).

We apply this model to the data in Table 1.1, which gives the numbers of pump failures, Y_i, observed in t_i thousands of hours for $k = 10$ different systems of a certain nuclear power plant. The observations are listed in increasing order of raw failure rate $r_i = Y_i/t_i$, the classical point estimate of the true failure rate θ_i for the ith system. These data (and the corresponding `WinBUGS` code in Figure 1.4) are also available within `WinBUGS`: simply click on `Help`, pull down to `Examples Vol I` and see the second example in the list.

The full conditional distributions for the θ_i and β are available in closed form (as gamma distributions), but the full conditional distribution for α is not standard. However, its form is

$$p(\alpha|\beta,\{\theta_i\},\mathbf{y}) \propto \left[\prod_{i=1}^{k} g(\theta_i|\alpha,\beta)\right] h(\alpha)$$
$$\propto \left[\prod_{i=1}^{k} \frac{\theta_i^{\alpha-1}}{\Gamma(\alpha)\beta^{\alpha}}\right] e^{-\alpha/\mu}$$

```
model
{
  for (i in 1:k) {
    theta[i] ~ dgamma(alpha,beta)
    lambda[i] <- theta[i]*t[i]
    Y[i] ~ dpois(lambda[i])
  }
  alpha ~ dexp(1.0)
  beta ~ dgamma(0.1, 1.0)
}

DATA:
list(k = 10, Y = c(5, 1, 5, 14, 3,
  19, 1, 1, 4, 22),
  t = c(94.320, 15.72, 62.88,
      125.76, 5.24, 31.44, 1.048,
      1.048, 2.096, 10.48))

INITS:
list(theta=c(1,1,1,1,1,1,1,1,1,1),
  alpha=1, beta=1) }
```

Figure 1.4. WinBUGS code for the pump data example.

which can be shown to be log-concave in α, so that `WinBUGS` may use adaptive rejection sampling here. For posteriors for which log-concavity cannot be readily checked, `WinBUGS` uses Metropolis sampling with a Gaussian proposal density.

We choose the values $\mu = 1$, $c = 0.1$, and $d = 1.0$, resulting in reasonably vague hyperpriors for α and β. Results from running 1000 burn-in samples, followed by a "production" run of 10,000 samples (single chain) are given in Table 1.2. Note that while θ_5 and θ_6 have very similar posterior means, the latter posterior is much narrower (i.e. smaller posterior standard deviation). This is because, while the crude failure rates for the two pumps are similar, the latter is based on a far greater number of hours of observation ($t_6 = 31.44$, while $t_5 = 5.24$). Hence we "know" more about pump 6, and this is properly reflected in its posterior distribution.

1.6.3 Bayesian kriging

As a third example, consider a point-level spatial (kriging) model of the form

$$\mathbf{Y} \sim \text{MVN}(\boldsymbol{\mu}, w^2 H(\phi) + v^2 I),$$

where $\mathbf{Y} = (Y(\mathbf{s}_1),\ldots,Y(\mathbf{s}_n))'$ is a vector of observations at spatial locations $\mathbf{s}_i, i = 1,\ldots,n$, and we

Table 1.2 Posterior summaries, Pump data model

node	mean	sd	MC error	2.5%	median	97.5%
alpha	0.7001	0.2699	0.004706	0.2851	0.6634	1.338
beta	0.929	0.5325	0.00978	0.1938	0.8315	2.205
theta[1]	0.0598	0.02542	2.68E-4	0.02128	0.05627	0.1195
theta[5]	0.6056	0.315	0.003087	0.1529	0.5529	1.359
theta[6]	0.6105	0.1393	0.0014	0.3668	0.5996	0.9096
theta[10]	1.993	0.4251	0.004915	1.264	1.958	2.916

```
model
{
for(i in 1:N) {
  Y[i] ~ dnorm(mu[i], tauv)
  mu[i] <- inprod(X[i,],beta[]) + W[i]
  muW[i] <- 0
  }
for(i in 1:p) {beta[i] ~ dnorm(0.0,
  0.0001)}
W[1:N] ~ dmnorm(muW[], Omega[,])
tauv ~ dgamma(0.001,0.001)
v <- 1/sqrt(tauv)
tauw ~ dgamma(0.001,0.001)
w <- 1/sqrt(tauw)
phi ~ dgamma(0.01,0.01)

for (i in 1:N){
  for(j in 1:N){
    H[i,j] <- (1/tauw)
              *exp(-phi*pow(d[i,j],2))}}
Omega[1:N,1:N] <- inverse(H[1:N,1:N])
}
```

Figure 1.5. WinBUGS code for the Bayesian kriging example using only standard functions.

```
model
{
for(i in 1:N) {
  Y[i] ~ dnorm(mu[i], tauv)
  mu[i] <- inprod(X[i,],beta[]) + W[i]
  muW[i] <- 0
  }
for(i in 1:p) {beta[i] ~ dnorm(0.0,
                0.0001)}
W[1:N] ~ spatial.exp(muW[], x[], y[],
          tauw, phi, 1)
tauv ~ dgamma(0.001,0.001)
v <- 1/sqrt(tauv)
tauw ~ dgamma(0.001,0.001)
w <- 1/sqrt(tauw)
phi ~ dgamma(0.01,0.01)
}
```

Figure 1.6. WinBUGS code for the Bayesian kriging example using the `spatial.exp` function.

assume $\boldsymbol{\mu} = X\boldsymbol{\beta}$ where the design matrix X will also likely depend on location. Here, σ^2 is the variance of the spatial model (or partial sill) and τ^2 is the pure error variance (or nugget) so that $\sigma^2 + \tau^2$ is the sill. I is an $n \times n$ identity matrix, while $\Sigma = w^2 H(\phi)$, an $n \times n$ correlation matrix having exponential form $H(\phi)_{ij} = \exp(-\phi d_{ij})$, where d_{ij} is the distance between locations i and j.

Figure 1.5 gives `WinBUGS` code to do this problem directly, that is, using the multivariate normal distribution `dmnorm` and constructing the H matrix explicitly using the exponential (`exp`) and power (`pow`) functions, the distances d_{ij}, and the partial sill and range parameters τ_w and ϕ. This H is then inverted to give Ω, the precision of the random effects W_i. Finally, the W_i are added into the mean structure created via the `inprod` command, with the nugget precision τ_v incorporated into the normal distribution of the data itself.

This kriging model can be handled in a better way using the `spatial.exp` function now available in `WinBUGS` releases 1.4 and later; this code is given in Figure 1.6. Note the `spatial.exp` function simplifies the specification of the spatial random effects W_i (where the lower case `x` and `y` refer to the x and y coordinates of the spatial locations $\mathbf{s}_i$), but the nugget term τ_v must still be added separately. Finally, this code handles spatial estimation, but for prediction of unseen values $\mathbf{Y}_0$ at new sites having covariate values $\mathbf{X}_0$, we would add in a loop utilizing the `spatial.pred` function; for details see the `WinBUGS` spatial help (click on `Map`

and pull down to `Manual`) or Banerjee et al. (2004, Section 5.1). The source code for Figures 1.3 and 1.4 is taken from the website `http://www.biostat.umn.edu/~brad/ph8436.html`.

1.7 Summary

In summary, this chapter has attempted a broad overview of many different topics. We have asserted a general modeling formulation for ecological processes. We have shown that such a formulation leads us to hierarchical modeling and, more generally, to graphical modeling. We have argued that fitting and inference for such models is most naturally implemented within the Bayesian framework. We have briefly reviewed the issues in Bayesian inference and Bayesian model comparison. Finally, we have noted that simulation-based model fitting (in the form of MCMC) is a valuable tool for carrying out the fitting and inference. Recognizing that we have offered really a "bare bones" exposure to all of this material, we strongly encourage the reader to look further into the literature and we have attempted to provide a bibliography suitable to do so.

References

Chapter 1

Agarwal, D. K. and A. E. Gelfand. 2005. Slice Gibbs sampling for simulation based fitting of spatial data models. *Statistics and Computing* (forthcoming)

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. In: Petrov and Czaki (eds), *Proceedings of the 2nd International Symposium on Information Theory*, pp. 267–281.

Banerjee, S., B. P. Carlin, and A. E. Gelfand. 2004. *Hierarchical Modeling and Analysis for Spatial Data*, Chapman and Hall/CRC Press, Boca Raton, FL.

Berger, J. O. 1985. *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. Springer-Verlag, New York.

Bernardo, J. M. and A. F. M. Smith. 1994. *Bayesian Theory*. Wiley, New York (with discussion).

Carlin, B. P. and T. A. Louis. 2000. *Bayes and Empirical Bayes Methods for Data Analysis*, 2nd ed. Chapman and Hall/CRC Press, Boca Raton, FL.

Chen, M.-H., Q.-M. Shao, and J. G. Ibrahim. 2000. *Monte Carlo Methods in Bayesian Computation*. Springer-Verlag, New York.

Clark, J. S. 2005. *Models for Ecological Data: Statistical Computation for Classical and Bayesian Frameworks*. Princeton University Press, Princeton, NJ (in press).

Congdon, P. 2001. *Bayesian Statistical Modelling*. Wiley, Chichester.

Cowell, R. G., A. P. Dawid, S. Lauritzen, and D. J. Spiegelhalter. 1999. *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York.

Cowles, M. K. and B. P. Carlin. 1996. Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association* **91**: 883–904.

Cressie, N. A. C. 1993. *Statistics for Spatial Data*, 2nd ed. Wiley, New York.

Damien, P., J. Wakefield, and S. Walker. 1999. Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society, Series B* **61**: 331–344.

DeGroot, M. H. 1970. *Optimal Statistical Decisions*. McGraw-Hill, New York.

Gamerman, D. 1997. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman and Hall/CRC Press, Boca Raton, FL.

Gaver, D. P. and I. G. O'Muircheartaigh. 1987. Robust empirical Bayes analyses of event rates. *Technometrics* **29**: 1–15.

Gelfand, A. E. and A. F. M. Smith. 1990. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* **85**: 398–409.

Gelfand, A. E. and S. K. Ghosh. 1998. Model choice: a minimum posterior predictive loss approach. *Biometrika* **85**: 1–11.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin. 2004. *Bayesian Data Analysis*, 2nd ed. Chapman and Hall/CRC Press, Boca Raton, FL.

Gelman, A., G. O. Roberts, and W. R. Gilks. 1996. Efficient Metropolis jumping rules. In: J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (eds), *Bayesian Statistics 5*. Oxford University Press, Oxford, pp. 599–607.

Gelman, A. and D. B. Rubin. 1992. Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science* **7**: 457–511.

Geman, S. and D. Geman. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**: 721–741.

Geyer, C. J. 1992. Practical Markov Chain Monte Carlo (with discussion). *Statistical Science* **7**: 473–511.

Gilks, W. R. and P. Wild. 1992. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society, Series C (Applied Statistics)* **41**: 337–348.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**: 97–109.

AQ: Please update ref. Agarwal and Gelfand 2005.

AQ: Please update ref. Clark 2005.

Kass, R. E. and A. E. Raftery. 1995. Bayes factors. *Journal of the American Statistical Association* **90**: 773–795.

Lee, P. M. 1997. *Bayesian Statistics: An Introduction*, 2nd ed. Arnold, London.

Liu, J. S. 2001. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York.

Maritz, J. S. and T. Lwin. 1989. *Empirical Bayes Methods*, Chapman Hall, London.

Mengersen, K. L., C. P. Robert, and C. Guihenneuc-Jouyaux. 1999. MCMC convergence diagnostics: a review (with discussion). In: J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith. (eds), *Bayesian Statistics 6*. Oxford University Press, Oxford, pp. 415–440.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equations of state calculations by fast computing machines. *Journal of Chemical Physics* **21**: 1087–1091.

O'Hagan, A. 1994. *Kendall's Advanced Theory of Statistics Volume 2b: Bayesian Inference*. Edward Arnold, London.

Robert, C. P. 1994. *The Bayesian Choice: A Decision-Theoretic Motivation*. Springer-Verlag, New York.

Robert, C. P. and G. Casella. 1999. *Monte Carlo Statistical Methods*. Springer-Verlag, New York.

Roberts, G. O. and A. F. M. Smith. 1993. Simple conditions for the convergence of the Gibbs sampler and Metropolis–Hastings algorithms. *Stochastic Processes and their Applications* **49**: 207–216.

Schervish, M. J. and B. P. Carlin. 1992. On the convergence of successive substitution sampling. *Journal of Computational and Graphical Statistics* **1**: 111–127.

Spiegelhalter, D. J., N. Best, B. P. Carlin, and A. van der-Linde. 2002. Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B* **64**: 583–639.

Spiegelhalter, D. J., A. Thomas, N. Best, and W. R. Gilks. 1995*a*. BUGS: Bayesian inference using Gibbs sampling, Version 0.50. Technical report, Medical Research Council Biostatistics Unit, Institute of Public Health, Cambridge University.

Spiegelhalter, D. J., A. Thomas, N. Best, and W. R. Gilks. 1995*b*. BUGS examples, Version 0.50. Technical report, Medical Research Council Biostatistics Unit, Institute of Public Health, Cambridge University.

Tanner, M. A. 1996. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*, 3rd ed. Springer-Verlag, New York.

Tierney, L. 1994. Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics* **22**: 1701–1762.

Whittaker, J. 1990. *Graphical Models in Applied Multivariate Statistics*. Wiley, Chichester.

Zhu, L. and B. P. Carlin. 2000. Comparing hierarchical models for spatio-temporally misaligned data using the Deviance Information Criterion. *Statistics in Medicine* **19**: 2265–2278.

Chapter 2

Akaike, H. 1973. Information theory as an extension of the maximum likelihood principle. In: B. N. Petrov and F. Csaki (eds), *Second International Symposium on Information Theory*. Akademiai Kiado, Budapest, pp. 267–281.

Bahlo, M. and R. C. Griffiths. 2001. Coalescence time for two genes from a subdivided population. *Journal of Mathematical Biology* **43**: 397–410.

Burnham, K. P. and D. R. Anderson. 2002. *Model Selection and Multimodel Inference: A Practical Information–Theoretic Approach*. Springer Verlag, New York.

Casella, G. and R. L. Berger. 2002. *Statistical Inference*, 2nd edn. Duxbury, Pacific Grove, CA.

Cockerham, C. C. 1969. Variance of gene frequencies. *Evolution* **23**: 72–84.

Crow, J. F. and K. Aoki. 1984. Group selection for a polygenic behavioral trait: estimating the degree of population subdivision. *Proceedings of the National Academy of Sciences USA* **81**: 6073–6077.

Crow, J. F. and M. Kimura. 1970. *An Introduction to Population Genetics Theory*. Burgess Publishing Company, Minneapolis, MN.

Ewens, W. J. 1979. *Mathematical Population Genetics*. Springer Verlag, Berlin.

Excoffier, L. 2001. Analysis of population subdivision. In: D. J. Balding, M. Bishop, and C. Cannings (eds), *Handbook of Statistical Genetics*. John Wiley and Sons, Chichester, pp. 271–307.

Fu, R., A. E. Gelfand, and K. E. Holsinger. 2003. Exact moment calculations for genetic models with migration, mutation, and drift. *Theoretical Population Biology* **63**: 231–243.

Gilks, W. R., S. Richardson, and D. J. Spiegelhalter. 1996. Introducing Markov chain Monte Carlo. In: W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (eds), *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, New York, pp. 1–19.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**: 97–109.