Markov Chain Monte Carlo Methods for Statistical Inference

Julian Besag[1]

Department of Statistics
University of Washington,
Seattle, USA

Spring 2004

SUMMARY

These notes provide an introduction to Markov chain Monte Carlo methods and their applications to both Bayesian and frequentist statistical inference. Such methods have revolutionized what can be achieved computationally, especially in the Bayesian paradigm. The account begins by discussing ordinary Monte Carlo methods: these have the same goals as the Markov chain versions but can only rarely be implemented. Subsequent sections describe basic Markov chain Monte Carlo, based on the Hastings algorithm and including both the Metropolis method and the Gibbs sampler as special cases, and go on to discuss some more specialized developments, including adaptive slice sampling, exact goodness–of–fit tests, maximum likelihood estimation, the Langevin–Hastings algorithm, auxiliary variables techniques, perfect sampling via coupling from the past, reversible jumps methods for target spaces of varying dimensions, and simulated annealing. Specimen applications are described throughout the notes.

Keywords: Adaptive slice sampling; Autologistic distribution; Auxiliary variables; Bayesian computation; Competing risks; Contingency tables; Darwin's finches; Exact p–values; Gibbs sampler; Hastings algorithm; Hidden Markov models; Importance sampling; Ising model; Langevin diffusion; Markov chain Monte Carlo; Markov random fields; Maximum likelihood estimation; Metropolis method; Mixture models; Noisy binary channel; Perfect simulation; Point processes; Rasch model; Reversibility; Reversible jumps; Simulated annealing; Spatial statistics; Swendsen–Wang algorithm; Traveling salesman; Weibull distribution; Wolff's algorithm.

[1] Address for correspondence: Department of Statistics, University of Washington, Box 354322, Seattle WA 98195, USA; E-mail: julian@stat.washington.edu

1 The computational challenge

1.1 Introduction

More than fifty years ago, Metropolis, Rosenbluth, Rosenbluth, Teller and Teller (1953) introduced the Metropolis algorithm into the physics literature. With the notable exception of Hammersley and Handscomb (1964, Ch. 9), there was little interest from statisticians in such Markov chain Monte Carlo (MCMC) methods, perhaps because they did not have easy access to "fast" computers. Then, in the 1970's, spatial statisticians, in part stimulated by Hastings (1970) and Hammersley and Clifford (1971), began experimenting with MCMC in the study of lattice systems and spatial point processes, both of which could be simulated via discrete or continuous time Markov chains. In the early 1980's, Donald and Stuart Geman forged a link between MCMC and digital image analysis, inspired in part by Ulf Grenander's work on general pattern theory and his maxim "Pattern analysis = Pattern synthesis" (Grenander, 1983). In particular, their seminal paper (Geman and Geman, 1984) adopts a Bayesian approach, with Markov random fields (e.g. Besag, 1974) as prior distributions, and either the Metropolis algorithm or the Gibbs sampler to synthesize the posterior distribution of image attributes; and it uses the closely related method of simulated annealing (Kirkpatrick, Gelatt and Vecchi, 1983) to determine an approximation to the most probable (MAP) image. Bayesian data analysis is particularly well suited to MCMC because it generally requires the evaluation of high–dimensional integrals that are not amenable to more conventional numerical methods.

The inclusion of hyperparameters for a fully Bayesian analysis, in the context of complex models in geographical epidemiology, was proposed in Besag (1989), following a suggestion by David Clayton, and was illustrated subsequently in Besag, York and Mollié (1991). However, the two papers that caught the attention of the Bayesian community were those by Gelfand and Smith (1990) and Gelfand, Hills, Racine–Poon and Smith (1990). These set the stage for a revolution in Bayesian data analysis, with MCMC, initially in the guise of the Gibbs sampler, quickly becoming the standard computational workhorse. MCMC has brought about something of a reversal of fortunes and it is now routine to fit extremely complex formulations in the Bayesian paradigm that still seem out of reach of frequentist methods. Indeed, the limiting factor in complexity is more often the lack of appropriate data than the inability to perform the computations.

Although its impact has been much less than in the Bayesian paradigm, MCMC also has an important and developing role in frequentist inference. Thus, MCMC maximum likelihood was rediscovered and generalized by Geyer (1991) and Geyer and Thompson (1992), following earlier work in spatial statistics by Penttinen (1984). The key idea here is to use importance sampling to extend the range of MCMC from the synthesis of a single target distribution to that of a family of distibutions that are in its vicinity. For example, this facilitates maximum likelihood estimation in complex generalized mixed models. The same trick is important in Bayesian sensitivity analysis.

Finally, the calculation of MCMC exact p–values (Besag and Clifford, 1989, 1991) has seen a recent revival, focusing on conditional tests in logistic regression, in multidimensional contingency tables, in Markov chains and in other applications for which standard asymptotic chi–squared theory breaks down.

The intention of this tutorial is to provide an introduction to MCMC methods in statistical inference. Other descriptions can be found in the books by (or edited by) Gelman, Carlin, Stern and Rubin (1995), Fishman (1996), Gilks, Richardson and Spiegelhalter (1996), Gamerman (1997), Robert and Casella (1999), Chen, Shao and Ibrahim (2000), Doucet, de Freitas and Gordon (2001), Liu (2001), MacCormick (2002) and Winkler (2003). Some early papers that contain review material include Geyer (1992), Smith and Roberts (1993), Besag and Green (1993), Tierney (1994) and Besag, Green, Higdon and Mengersen (1995).

In the remainder of this section, we provide a list of topics to be covered and also describe the main computational task. Thus, Section 2 is devoted to ordinary Monte Carlo methods and their relevance, at least in principle, to both Bayesian and frequentist inference. Specifically, Section 2.1 is concerned with Bayesian computation, exemplified by the analysis of hidden Markov models and the special case of the noisy binary channel. In addition to their inherent practical importance, hidden Markov models lie at the interface between what can be achieved using ordinary Monte Carlo and MCMC. Section 2.2 describes exact frequentist Monte Carlo p–values, exemplified by simple tests of independence in (sparse) two– and higher–dimensional contingency tables and contrasted with the difficulties of simulating from the Rasch model. Sections 2.3 and 2.4 discuss importance sampling and its relevance to Monte Carlo maximum likelihood estimation, illustrated by fitting a conditional Ising model to the initial pattern of disease in an array of endive plants. Section 2.5 describes a toy version of simulated annealing as a method of discrete optimization.

Unfortunately, the implementation of ordinary Monte Carlo methods is rarely feasible in practice, except for the types of rather simplistic problems considered in Section 2. Nevertheless, the underlying ideas transfer seamlessly to MCMC, as described in Section 3.1, with random samples replaced by dependent samples from a Markov chain. Sections 3.2 and 3.3 discuss the detailed balance condition for stationary distributions and how it relates to the remarkably simple Hastings construction that drives almost all MCMC algorithms in statistical applications. Sections 3.4 to 3.7 focus on basic algorithms, including the single–component Gibbs sampler and Metropolis method. Sections 3.8 and 3.9 provide contrasting examples, one in curve fitting that is well suited to (block) Gibbs, the other on the poly–Weibull distribution for competing risks in survival analysis that is not.

Section 4 discusses some more specialized topics and some additional applications. Each subsection is self–contained and can be read in virtual isolation from the rest. Thus, Section 4.1 describes adaptive slice sampling (Neal, 2003), which provides an alternative to Gibbs in writing general software for Bayesian inference.

Then Section 4.2 describes exact frequentist MCMC p–values for goodness of fit. The device of conditioning on sufficient statistics, so as to eliminate unknown parameters from the target distribution, requires the construction of constrained MCMC algorithms, a topic that is currently of much interest. Our first example involves the Rasch model and the data on species diversity previously encountered in Section 2.2.3; the second revisits the Ising model and the pattern of disease among the endives.

Section 4.3 returns to the endives data to exemplify MCMC maximum likelihood estimation but there is little to add to Section 2.4. Section 4.4 is devoted to the Langevin–Hastings algorithm; neither of our toy examples is persuasive but we refer elsewhere for a more complex and convincing application in point processes.

Section 4.5 describes auxiliary variables methods and the Swendsen–Wang algorithm applied to the autologistic distribution, with the Ising model as a special case. We also pay a return visit to the Bayesian analysis of competing risks.

Section 4.6 discusses perfect random sampling via MCMC, which at first may seem a contradiction. In particular, we describe Propp and Wilson's coupling from the past, using the noisy binary channel of Section 2.1.1 and implicitly the autologistic distribution as illustrations. Indeed, our implementation of maximum likelihood estimation for the Ising model in Section 2.4.1 relies on perfect MCMC and is really a contrived example of ordinary Monte Carlo.

Section 4.7 is concerned with the widely used reversible jumps algorithm introduced by Green (1995). We provide an alternative description and, as an illustration, return once more to competing risks and the poly–Weibull distribution, now allowing an unknown number of components in the mixture.

Section 4.8 applies simulated annealing to a simple traveling salesman problem. Even though this is not in itself a task in statistical inference, it is easy to describe and provides a template for discrete optimization problems in decision theory, such as determining the maximum a posteriori (MAP) estimate in Bayesian image analysis. Also, simulated annealing can be adapted to corresponding continuum tasks.

The presentation in these notes differs from most others in providing a somewhat unified description of how MCMC methods relate to both Bayesian and frequentist inference. Numerical examples are drawn from a range of disciplines but, as regards complex Bayesian analyses, there is already an abundance in the literature and it is easy to find applications that match one's own personal interests. Spatial topics in which the author has been involved but that are not discussed here include geographical epidemiology (Besag *et al.*, 1991; Knorr–Held and Besag, 1998; Byers and Besag, 2000), agricultural field experiments (e.g. Besag and Higdon, 1999) and digital image analysis (e.g. Moffett, Besag, Byers and Li, 1997; Maitra and Besag, 1998). Nevertheless the notes certainly reflect the writer's own experiences, as anyone thumbing through the list of references will no doubt conclude!

Finally, we admit that this account provides little more than an introduction to MCMC in statistical inference. It has evolved over the past several years and, in some respects, has not succeeded in keeping pace with all the most recent developments. For the latest in technical reports, the reader should consult the MCMC web site at

`http://www.statslab.cam.ac.uk/∼mcmc/`

Perhaps the most glaring omission here is discussion of sequential MCMC, also referred to as particle filters, despite its extensive use in the dynamic analysis of time series, ranging from financial data to target tracking. Books that focus on sequential MCMC include Doucet *et al.* (2001) and MacCormick and McCormick (2002); and Liu (2001) also deals extensively with the topic.

1.2 The main task

Let X denote a random quantity: in practice, X will have many components and might represent, for example, a random vector or a multi–way contingency table or a grey–level pixel image (perhaps augmented by other variables). Additionally, some components of X may be discrete and others continuous. However, it is most convenient for the moment to think of X as a single random variable (r.v.), having a finite but extremely complicated sample space. Indeed, in a sense, such a formulation is perfectly general because ultimately all our calculations will be made on a finite machine. It is only in describing quite specific MCMC algorithms, such as the Gibbs sampler or the adaptive slice sampler, that we need to address the individual components of X.

Thus, let $\{\pi(x) : x \in S\}$ denote the probability mass function (p.m.f.) of X, where S is the corresponding *minimal* sample space; that is, $S = \{x : \pi(x) > 0\}$. We assume that $\pi(.)$ is known up to scale, so that

$$\pi(x) \;=\; h(x)/c, \qquad x \in S, \tag{1}$$

where $h(.)$ is completely specified. In practice, the normalizing constant

$$c \;=\; \sum_{x \in S} h(x) \tag{2}$$

is rarely known in closed form and typically the space S is too large for c to be calculated directly from (2). Nevertheless, our goal is to compute expectations of particular functions g under π; that is, we require

$$\mathrm{E}_\pi g \;=\; \sum_{x \in S} g(x)\pi(x), \tag{3}$$

for any relevant g. Again, we assume that the summation in (3) cannot be carried out directly (even in the rare event that c is known).

As an especially important special case, note that (3) includes the *probability* of any particular event concerning X. Explicitly, for any relevant subset B of S,

$$\Pr(X \in B) \;=\; \sum_{x \in S} 1[x \in B]\,\pi(x), \tag{4}$$

where $1[\,.\,]$ is the usual indicator function; that is, $1[x \in B] = 1$ if the outcome x implies that the event B occurs and $1[x \in B] = 0$ otherwise. One of the major strengths of MCMC is its ability to focus directly on probabilities, in contrast to the more usual tradition of indirect calculation via moment approximations and limit theorems.

2 Ordinary Monte Carlo calculations

As suggested in Section 1, it is convenient to introduce the underlying aims of MCMC by first describing ordinary Monte Carlo calculations. Thus, for the moment, we suppose that, despite the complexity of S, we are able to generate random draws $x^{(1)}, x^{(2)}, \ldots$ from the target distribution π, corresponding to independent and identically distributed (i.i.d.) r.v.'s $X^{(1)}, X^{(2)}, \ldots$. If we produce m such draws, $x^{(1)}, \ldots, x^{(m)}$, then the obvious estimate of $\mathrm{E}_\pi g$ is the empirical mean,

$$\bar{g} \;=\; \frac{1}{m} \sum_{t=1}^{m} g(x^{(t)}). \tag{5}$$

The superscript notation $x^{(t)}$ is rather clumsy but we prefer to reserve subscripts for later, when it becomes necessary to recognize explicitly that x is a vector or table or whatever and to refer to its individual components.

Of course, $\bar{g}$ is an unbiased estimate of $\mathrm{E}_\pi g$ and has a sampling distribution that is approximately Gaussian, with variance σ^2/m, where σ^2 can be estimated by

$$s^2 \;=\; \frac{1}{m-1} \sum_{t=1}^{m} \{g(x^{(t)}) - \bar{g}\}^2, \tag{6}$$

assuming appropriate regularity conditions. Thus, point and interval estimates for $\mathrm{E}_\pi g$ can be constructed in the usual way. When $g(x) = 1[x \in B]$ and we are concerned with a probability (4), interval estimates can be sharpened in the usual way via the underlying binomial distribution.

Thinking ahead, we note that sometimes (5) provides a valid approximation to $\mathrm{E}_\pi g$ even when $x^{(1)}, \ldots, x^{(m)}$ do not form a random sample from π. In particular, this is so when m is sufficiently large and $X^{(1)}, X^{(2)}, \ldots$, seeded by some $x^{(0)} \in S$, form an ergodic (here regular) Markov chain with (finite) state space S and limiting distribution π. This extension provides the basis for MCMC and is required when random sampling from π is no longer feasible. It assumes that useful recipes exist for constructing appropriate transition probability matrices, an assumption that we verify in due course. However, for the moment, we avoid any complications caused by possible dependence among the r.v.'s $X^{(1)}, X^{(2)}, \ldots$, including modifications to the sampling theory in the previous paragraph, and assume that random samples from π are indeed available. In this rather artificial setting, we follow the schedule in Section 1.1 and discuss how ordinary Monte Carlo sampling relates to both Bayesian and frequentist statistical inference. We include some illustrative examples and also comment in passing on the limitations of simple Monte Carlo methods and on the corresponding role of MCMC.

2.1 Bayesian computation

The above brief description of ordinary Monte Carlo calculations is presented in a frequentist framework but the idea itself transfers immediately to (parametric) Bayesian inference. Thus, let x now denote an unknown (scalar) parameter in a finite parameter space S and suppose that $\{\rho(x) : x \in S\}$ is a prior p.m.f. representing our initial beliefs about the true value of x. Let y denote relevant data, with corresponding known likelihood $L(y|x)$, so that the posterior p.m.f. for x given y is

$$\pi(x|y) \;\propto\; L(y|x)\rho(x), \qquad\qquad x \in S. \tag{7}$$

In terms of equations (1) and (2), we replace $\pi(x)$ by $\pi(x|y)$, with

$$h(x) \;\propto\; L(y|x)\rho(x); \tag{8}$$

c is the associated (unknown) normalizing constant. Recall that, in the Bayesian paradigm, inferences are conditional on the fixed data y. Note that we have written proportionality in (8), in case $L(y|x)$ and $\rho(x)$ are known only up to scale.

Now suppose that $x^{(1)}, \ldots, x^{(m)}$ is a large random sample from $\pi(x|y)$ for fixed y. Then, we can use (5) to approximate $\mathrm{E}_\pi g$, the posterior mean of g, for any particular g. For example, we can evaluate posterior probabilities concerning the parameter x and construct corresponding credible intervals. The approach is essentially unchanged if the parameter space S is continuous rather than discrete. Further, it extends immediately to multi–component parameters, though, in practice, it is usually very difficult or impossible to sample directly from a multivariate π, in which case we must resort to MCMC.

It is perhaps worth emphasizing that the availability of random samples from $\pi(x|y)$ would permit trivial solutions to traditionally very complicated problems. For example, consider a clinical, industrial or agricultural trial in which the aim is to compare different treatment effects θ_i. Then $x = (\theta, \phi)$, where θ is the vector of θ_i's and ϕ is a vector of other, possibly uninteresting, parameters in the posterior distribution. A natural quantity of interest from a Bayesian perspective is the posterior probability that any particular treatment effect is best or is among the best three, say, where here we suppose best to mean having the largest effect. Such demands are usually far beyond the capabilities of conventional numerical methods, because they involve summations (or integrations) of non–standard functions over awkward regions of the parameter space S. However, in the present context, we can closely approximate the probability that treatment i is best, simply by the proportion of simulated $\theta^{(t)}$'s among which $\theta_i^{(t)}$ is the largest component; and the probability that treatment i is one of the best three by the proportion of $\theta^{(t)}$'s for which $\theta_i^{(t)}$ is one of the largest three components. Incidentally, note that the extremely unsatisfactory issues that occur in a frequentist setting when treatment i is selected in the light of the data do not arise in the Bayesian paradigm.

Ranking and selection is just one area in which the availability of random samples from posterior distributions would have had a profound influence on applied Bayesian inference.

Not only does MCMC deliver what ordinary Monte Carlo methods have failed to achieve but, in addition, MCMC encourages the data analyst to build and analyze more realistic statistical models that may be far more complex than standard formulations. Indeed, one must often resist the temptation to build representations whose complexity cannot be justified by the underlying scientific problem or by the available data!

2.1.1 Hidden Markov models

Although ordinary Monte Carlo methods can rarely be implemented in Bayesian inference, hidden Markov chains provide an exception, at least in a simplified version of the general problem. Although a Markov chain is involved, this arises as an ingredient of the original model, specifically in the prior distribution for the unobserved (hidden) output sequence from the chain, and not merely as a computational device. The posterior distribution retains the Markov property, conditional on the data, and can be simulated via the Baum algorithm (Baum, Petrie, Soules and Weiss, 1970), though below we adopt an alternative backwards recursion (Bartolucci and Besag, 2002) that also requires $O(n)$ flops for a sequence of length n. Applications of hidden Markov models occur in speech recognition (e.g. Rabiner, 1989; Juang and Rabiner, 1991), in neurophysiology (e.g. Fredkin and Rice, 1992), in computational biology (e.g. Haussler, Krogh, Mian and Sjolander, 1993; Eddie, Mitchison and Durbin, 1995; Liu, Neuwald and Lawrence, 1995), in climatology (e.g. Hughes, Guttorp and Charles, 1999), in epidemiologic surveillance (Le Strat and Carrat, 1999) and elsewhere; see also MacDonald and Zucchini (1997).

To describe a hidden Markov chain, let $x = (x_1, \dots, x_n)$ be the output sequence from some process, where $x \in \{0, 1, \dots, s\}^n$. We suppose that the signal x cannot be observed but that each unknown x_i generates an observation y_i with known probability $f(x_i, y_i)$. We assume conditional independence, so that the probability of $y = (y_1, \dots, y_n)$, given x, is

$$L(y|x) \;=\; \prod_{i=1}^{n} f(x_i, y_i). \tag{9}$$

Our goal is to make inferences about the unknown x from the data y. Of course, the obvious point estimate is $\breve{x} = \arg\max_x L(y|x)$ but suppose that we possess the additional information that x can be represented as the output from a stationary ergodic Markov chain, with known transition probability $q(x_i, x_{i+1})$ of the ith component x_i being followed by x_{i+1}. That is, x has marginal probability,

$$\rho(x) \;=\; q(x_1) \prod_{i=1}^{n-1} q(x_i, x_{i+1}), \tag{10}$$

where $q(.)$ is the stationary distribution implied by $q(.,.)$. If we regard $\rho(x)$ as a prior p.m.f.

for x, then the corresponding posterior probability of x, given y, is

$$\pi(x|y) \;\propto\; q(x_1)f(x_1,y_1)\prod_{i=2}^{n} q(x_{i-1},x_i)f(x_i,y_i). \tag{11}$$

If we can generate a random sample of signals $x^{(1)},\ldots,x^{(m)}$ from $\pi(x)=\pi(x|y)$, for fixed y, then we can use these to make inferences about the true x. However, the distribution defined by (11) is awkward to deal with, especially when n is very large, as is typical in applications.

At this point, we comment briefly on the practical relevance of the above specification. First, if the x_i's are truly generated by a Markov chain with known transition probabilities, then nothing intrinsically Bayesian arises in the formulation. Also, the Baum algorithm can be applied directly to evaluate most expectations (3) of interest, without recourse to random sampling, and even to determine $x^+ = \arg\max_x \pi(x)$, the MAP (maximum a posteriori) estimate of x, via the Viterbi algorithm. Second, if $\rho(.)$ is merely a representation of our beliefs about x, then we should also include uncertainty about the transition probabilities in the prior; and, in that case, random sampling from the posterior is no longer feasible. Despite these reservations, the description here is not only of academic interest, because fully Bayesian formulations can be tackled using an extension to MCMC of the random sampling algorithm discussed below; see Robert, Rydén and Titterington (2000).

Efficient algorithms for (11) depend on the fact that x given y inherits the Markov property, though its transition probabilities are functions of y and therefore non–homogeneous. Specifically,

$$\pi(x|y) \;=\; \pi(x_1|y)\prod_{i=2}^{n}\pi(x_i|x_{i-1},y), \tag{12}$$

where y in the final term can be replaced by $y_{\geq i}=(y_i,\ldots,y_n)$, a form of notation we adopt extensively. Although it is not necessary for the disinclined reader to work through the details below, we have included them because hidden Markov chains appear again in later examples and also similar conditional probability manipulations often arise in formulating MCMC algorithms. To establish (12), note that

$$\pi(x_{\geq k}|x_{<k},y) \;=\; \pi(x|y)\,/\,\pi(x_{<k}|y) \;\propto\; \pi(x|y), \tag{13}$$

because the denominator can be absorbed into the normalizing constant. Hence, (11) implies that

$$\pi(x_{\geq k}|x_{<k},y) \;\propto\; \prod_{i=k}^{n} q(x_{i-1},x_i)f(x_i,y_i), \qquad k=2,\ldots,n, \tag{14}$$

because terms in the product that involve only $x_{<k}$ and y can again be absorbed by the normalizing constant. The right–hand side of (14) depends only on x_{k-1} among $x_{<k}$, which is the Markov property. Also, (14) implies that $\pi(x_k|x_{<k},y)$ does not depend on $y_{<k}$. Incidentally, simple conditional probability results, typified by (13), are crucial in implementing MCMC, as we shall see later.

However, there is still a problem, because direct calculation of the transition probability $\pi(x_k|x_{k-1}, y_{\geq k})$ demands that we sum (13) over all $x_{k+1}, \ldots, x_n$ and clearly this is prohibitive in general. The Bartolucci and Besag (2002) algorithm depends on the following trivial but rather odd–looking result.

Lemma. Let U, V and W denote discrete r.v.'s. Then

$$\Pr(u|v) \;=\; \{\sum_w \Pr(w|u,v) \,/\, \Pr(u|v,w)\,\}^{-1}_{,}$$

provided the conditional probabilities are well defined.

If, in the lemma, we condition on y throughout, let $U = X_i$, $V = X_{i-1}$ and $W = X_{i+1}$, and apply the Markov property to $\pi(x_{i+1}|x_{i-1}, x_i, y)$, we obtain the backwards recursion,

$$\pi(x_i|x_{i-1}, y) \;=\; \{\sum_{x_{i+1}} \pi(x_{i+1}|x_i, y) \,/\, \pi(x_i|x_{i-1}, x_{i+1}, y)\,\}^{-1}_{,} \qquad i = n-1, \ldots, 1, \qquad (15)$$

with x_0 omitted when $n = 1$. The denominators in the summations can be evaluated explicitly because they depend on y only through y_i and so

$$\pi(x_i|x_{i-1}, x_{i+1}, y_i) \;\propto\; f(y_i|x_i)\, p(x_{i+1}|x_i)\, p(x_i|x_{i-1}), \qquad i = 1, \ldots, n-1, \qquad (16)$$

again omitting x_0 throughout when $n = 1$. The normalizing constants in (16) can be found by summing the right–hand sides over x_i. Hence, (15) can be used successively for $i = n-1, \ldots, 1$ to obtain the left–hand sides and these can be substituted successively for $i = 1, \ldots, n$ to simulate from each in turn to generate the sequence x. Note that, unlike the Baum algorithm, dummy normalizations are not required to combat numerical problems.

Ex. Noisy binary channel

The noisy binary channel provides the simplest example of a hidden Markov chain. Thus, suppose that both the hidden x_i's and the observed y_i's are binary and that the log–odds of correct to incorrect transmission of x_i to y_i are α, for each i independently, where α is known. Then the naive estimate of x is y if $\alpha > 0$, $1-y$ if $\alpha < 0$, and indeterminate if $\alpha = 0$. Now suppose the x_i's follow a stationary Markov chain, in which the transition probability matrix is symmetric, with known log–odds β in favor of $x_{i+1} = x_i$. The symmetries are merely for convenience and could easily be dropped but imply that $q(0) = q(1) = \frac{1}{2}$ in (10). The posterior probability (11) of a true signal x given data y reduces to

$$\pi(x|y) \;\propto\; \exp\,(\alpha \sum_{i=1}^{n} 1[x_i = y_i] + \beta \sum_{i=1}^{n-1} 1[x_i = x_{i+1}]), \qquad x \in S = \{0,1\}^n, \qquad (17)$$

where again $1[\,.\,]$ denotes the usual indicator function.

As a numerical illustration, we take $\alpha = \ln 4$, corresponding to a corruption probability $\frac{1}{5}$, and $\beta = \ln 3$, so that like follows like in the Markov chain with probability $\frac{3}{4}$. Now suppose we observe the record $y = 11101100000100010111$, so that $|S| = 2^{20} = 1048576$. For such a tiny state space, it is easy to calculate exact expectations by complete enumeration of the posterior distribution of x given y or by direct application of the Baum algorithm. However, here we apply the algorithm to generate a random sample of size 10000 from $\pi(x|y)$, which we use to estimate various expectations. Thus, we find $x_1 = 1$ in 8989 of the samples, suggesting a posterior probability of 0.899 versus the correct value 0.896; for $x_2 = 1$, we obtain 0.927 versus 0.924; and so on. Hence, the marginal posterior modes (MPM) estimate $\hat{x}$ is correctly identified as $\hat{x} = 11111100000000010111$; here, $\hat{x}_i$ is defined as the more probable of 0 and 1 in each position i, given y. Clearly, $\hat{x}$ is a smoothed version of the data, with two fewer isolated bits. The $\hat{x}_i$'s in positions $i = 4$, 12, 16 and 17 are the most doubtful, with estimated (exact) probabilities of $x_i = 1$ equal to 0.530 (0.541), 0.421 (0.425), 0.570 (0.570) and 0.434 (0.432). Although neither component 16 nor 17 flips in the MPM estimate, it is interesting that, if we examine them jointly, the probabilities of 00, 10, 01 and 11 are 0.362 (0.360), 0.203 (0.207), 0.068 (0.070) and 0.366 (0.362), respectively. Thus, there is a preference for 00 or 11, rather than the 10 obtained in $\hat{x}$.

The previous point about the MPM estimate emphasizes the fact that it is defined marginally for each component in turn and must not be confused with other criteria that involve joint distributions. Indeed, at the opposite extreme to MPM is the MAP estimate, the most probable configuration x^+, given y, which here is 11111100000000011111 or 11111100000000000111. It is easy to see that these two configurations have the same posterior probability, because each involves two unlike adjacencies and requires three elements to be corrupted in forming y. In our random sample, the two x^+'s are the most frequent configurations, occurring on 288 and 323 occasions, respectively, compared to the true probability 0.0304. Note that $\hat{x}$ and y itself occur 138 and 25 times, compared to the true probabilities 0.0135 and 0.0027. If one requires a single–shot estimate of the true signal, then the choice of a particular criterion, ultimately in the form of a loss function, should depend on the practical goals of the analysis. For example, the MAP estimate corresponds to zero loss for the correct x and unit loss for any incorrect estimate, regardless of the number of errors among its components; whereas MPM arises from an elementwise loss function and minimizes the expected total number of errors among all the components. A personal view is that a major benefit of a sampling approach is that it encourages one to investigate various aspects of the posterior distribution, rather than concentrating on a single criterion. However, note that sampling from the posterior is not generally suitable for finding the MAP estimate. In Section 2.5, we discuss the closely related method of *simulated annealing* (Kirkpatrick, Gelatt and Vecchi, 1983), which often performs quite successfully.

As a more taxing toy example, we apply the Baum algorithm to obtain a single realization x from a noisy binary chain, again with $\alpha = \ln 4$ and $\beta = \ln 3$ but now with $y = 1110011100\ldots$, a vector of length 100000, so that $|S| = 2^{100000}$. Then the maximum likelihood, MPM and MAP estimates of x all coincide with the data y. In the event, our random draw from $\pi(x|y)$ agrees with y in 77710 components. We return to this example subsequently in discussing both simulated annealing and coupling from the past.

Finally, we briefly consider some complications that can occur in practice. First, suppose that α and β are unknown parameters with prior distributions. Then, not only do we acquire additional terms from the new (continuous) priors but also there are terms in α and β that previously were irrelevant and that can no longer be ignored in the posterior $\pi(x,\alpha,\beta|y)$. Or suppose that x is a two–dimensional pixel image, in which 1's represent "object" pixels and 0's refer to "background". Then a Markov chain prior for x is no longer appropriate and might be replaced by a Markov random field with unknown parameters. Such complications and many others are not amenable to the approaches we have discussed here but can be tackled via MCMC to collect (dependent) samples from the corresponding posterior distribution and hence make valid inferences.

2.2 Frequentist goodness–of–fit tests

It is often necessary, particularly at a preliminary stage of data analysis, to investigate the compatibility between a known p.m.f. $\{\pi(x) : x \in S\}$ and a single observation $x^{(1)} \in S$. Recall here that when we talk about a "single observation", we may mean a vector or a table (as in the examples below) or an image or whatever. Also, our requirement that the distribution is known may have been achieved by conditioning on sufficient statistics to eliminate parameters from the original formulation (again, as in the examples below). In a frequentist analysis, evidence of a conflict between $x^{(1)}$ and π is usually quantified by a p–value obtained by comparing the observed value $u^{(1)}$ of a particular test statistic $u = u(x)$ with its distribution under π. Suppose here that large values of $u^{(1)}$ suggest a conflict, so that the p–value is the tail probability given by (3), with

$$g(x) \;=\; 1[u(x) \geq u^{(1)}]. \tag{18}$$

Note that, although there have been important advances in the production of software for such calculations, there are restrictions on the sizes of the datasets for which they can be used. Here, we assume that the summation cannot be evaluated directly but that, instead, it is possible to generate a random sample $x^{(2)}, \ldots, x^{(m)}$ from π, yielding values $u^{(2)}, \ldots, u^{(m)}$ of the test statistic. There are then two slightly different methods of constructing a p–value, though the distinction is sometimes blurred in the literature.

The more obvious of the two procedures is to approximate the tail probability, implicit in (3) and (18), by the proportion of simulated $x^{(t)}$'s for which $u^{(t)} \geq u^{(1)}$. This is the standard Monte Carlo approach. The estimate is usually accompanied by a confidence interval based on the binomial distribution. We now consider a less well–known construction.

2.2.1 Exact Monte Carlo p–values

A slight modification of the above estimation procedure produces an exact p–value (Dwass, 1957; Barnard, 1963). First, note that, if $x^{(1)}$ is from π, then, ignoring the possibility of ties, the rank of $u^{(1)}$ among $u^{(1)}, \ldots, u^{(m)}$ is uniform on $1, \ldots, m$. It follows that, if $u^{(1)}$ turns out to be kth largest among all m values, an exact p–value k/m can be declared. This modified procedure is referred to as a (simple) Monte Carlo test, though again we warn of some confusion in the literature between the two cases. The choice of m is governed largely by computational considerations, with $m = 99$ or 999 or 9999 the most popular. Note that, if several investigators carry out the same test on the same data x_1, they will generally obtain slightly different p–values, despite the fact that marginally each result is exact! Such differences should not be important at a preliminary stage of analysis and disparities diminish as m increases. Ties between ranks can occur with discrete data, in which case we quote a corresponding range of p–values. Alternatively, the problem can be eliminated by using a randomized rule. For detailed investigation of Monte Carlo tests when π corresponds to a random sample of n observations from a population, see Jöckel (1986) and especially Hall and Titterington (1989).

A useful refinement is provided by *sequential* Monte Carlo tests (Besag and Clifford, 1991). First, one specifies a maximum number of simulations $m - 1$, as before, but now additionally a minimum number h, typically 10 or 20. Then $x^{(2)}, \ldots, x^{(m)}$ are drawn sequentially from π but with the proviso that sampling is terminated if ever h of the corresponding $u^{(t)}$'s exceed $u^{(1)}$, in which case a p–value h/l is declared, where $l \leq m - 1$ is the number of simulations; otherwise, the eventual p–value is k/m, as before. To see that the p–value for early stopping is correct, we ignore the possibility of ties and define the r.v. L to be the number of simulations required for h exceedences. Then, if $L = l$, the p–value is given by the probability $\Pr(L \geq l)$ of an occurrence as or more extreme than l under π. Now $L \geq l$ implies that $L > l - 1$ or equivalently that the first observation in a sample of size l from π is among the h largest values. Hence, $\Pr(L \geq l) = h/l$, as required.

The advantage of sequential tests is that they can be designed to terminate usually very early when there is no evidence against π but to continue sampling and produce a finely graduated p–value when the evidence against the model is substantial. For example, if the model is correct and we choose $m = 1000$ and $h = 20$, the expected sample size is reduced to 98; see Besag and Clifford (1991) for further details.

Monte Carlo tests have been especially useful in the preliminary analysis of spatial data, where parameters can often be eliminated by conditioning on sufficient statistics; see, for example, Besag and Diggle (1977) and Diggle (1983). The simplest example occurs in assessing whether a spatial point pattern over a (perhaps awkwardly shaped) study region is consistent with a homogeneous Poisson process: by conditioning on the number of points, the test is reduced to one of uniformity. Below, we instead consider the simplest possible test in a contingency table.

2.2.2 Testing for independence in contingency tables

Let $x^{(1)}$ denote an observed $r \times s$ contingency table, having cells $\{(i, j) : i = 1, \ldots, r;$ $j = 1, \ldots, s\}$, and corresponding entries generated according to standard multinomial assumptions, with unknown probability θ_{ij} that any particular observation falls in cell (i, j). Our task is to assess whether such data are consistent with independence of row and column categorizations; that is, with

$$\theta_{ij} \;=\; \phi_i \, \psi_j, \tag{19}$$

where the ϕ_i's and ψ_j's represent unknown probability distributions.

Let X denote a random table with all the above characteristics and subject to the same row and column totals as $x^{(1)}$. Let x denote a corresponding observed table, with entries x_{ij}. Then the distribution π of X is a multivariate version of the hypergeometric distribution, in which the conditioning eliminates the ϕ_i's and the ψ_j's; specifically,

$$\pi(x) \;=\; \frac{\prod_i x_{i+}! \prod_j x_{+j}!}{x_{++}! \prod_i \prod_j x_{ij}!}, \qquad\qquad x \in S,$$

where S is the set of all tables having the same margins x_{i+} and x_{+j} as the original table $x^{(1)}$. It follows that π can be used as a reference distribution to calculate a p–value for $x^{(1)}$ using any particular test statistic $u(x)$. In principle, this can be carried out directly via equations (3) and (18) but the computations are not feasible except for rather small tables because S is much too large. Of course, if $u(x)$ is Pearson's X^2 or the deviance, we can apply the usual asymptotic chi–squared approximation but support for this breaks down in tables with a substantial proportion of low expected counts $x_{i+}x_{+j}/x_{++}$.

When exact computations and asymptotic results are inappropriate, we can turn instead to simple or sequential Monte Carlo tests. Patefield (1981) provides a convenient algorithm for generating samples from π and this also extends to tests of independence in higher dimensions, where problems of small expected values are more prevalent. Here we describe the algorithm in terms of a trivial 2×3 example, in which the data form the left–hand table below:

3	2	4
2	1	2

4	2	3
1	1	3

This is merely a frequency table formed from the original 14 observations: $(1, 1)$, $(1, 1)$, $(1, 1)$, $(1, 2)$, $(1, 2)$, $(1, 3)$, $(1, 3)$, $(1, 3)$, $(1, 3)$, $(2, 1)$, $(2, 1)$, $(2, 2)$, $(2, 3)$, $(2, 3)$, in some order. Conditioning on the margins, independence implies that there should be no association between the nine 1's and five 2's that occur as the first index and the five 1's, three 2's and six 3's that occur as the second. To generate a new table from the null distribution, all we need to do is to randomly permute the elements that appear as the second index with respect to the first. Thus, we might obtain new "observations" $(1, 2)$, $(1, 1)$, $(1, 3)$, $(1, 3)$, $(1, 3)$, $(1, 2)$, $(1, 1)$, $(1, 1)$, $(1, 1)$, $(2, 3)$, $(2, 3)$, $(2, 2)$, $(2, 3)$, $(2, 1)$, which result in the above right–hand table. We go through this procedure $m - 1$ times to obtain our Monte Carlo sample. For a three–way table, we would need to permute the second and third indices with respect to the first to generate each new table; and so on in higher dimensions.

Ex. Deaths by horsekicks in the Prussian Army

The 20×14 table below gives the number of deaths from horsekicks in the Prussian Army by year and corps (Bishop, Fienberg and Holland, 1975). It is well known that the 280 individual counts are consistent with a single Poisson distribution having mean 196/280. In working towards this conclusion, a preliminary test of row and column independence might be of interest but the expected counts are too small to rely on the standard chi–squared theory for X^2 or the deviance. Instead, we apply a simple Monte Carlo test with either test statistic to conclude that there is no conflict between the data and the hypothesis of independence.

Corps identifier

Year	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
1875	0	0	0	0	0	0	0	1	1	0	0	0	1	0	3
1876	2	0	0	0	1	0	0	0	0	0	0	0	1	1	5
1877	2	0	0	0	0	0	1	1	0	0	1	0	2	0	7
1878	1	2	2	1	1	0	0	0	0	0	1	0	1	0	9
1879	0	0	0	1	1	2	2	0	1	0	0	2	1	0	10
1880	0	3	2	1	1	1	0	0	0	2	1	4	3	0	18
1881	1	0	0	2	1	0	0	1	0	1	0	0	0	0	6
1882	1	2	0	0	0	0	1	0	1	1	2	1	4	1	14
1883	0	0	1	2	0	1	2	1	0	1	0	3	0	0	11
1884	3	0	1	0	0	0	0	1	0	0	2	0	1	1	9
1885	0	0	0	0	0	0	1	0	0	2	0	1	0	1	5
1886	2	1	0	0	1	1	1	0	0	1	0	1	3	0	11
1887	1	1	2	1	0	0	3	2	1	1	0	1	2	0	15
1888	0	1	1	0	0	1	1	0	0	0	0	1	1	0	6
1889	0	0	1	1	0	1	1	0	0	1	2	2	0	2	11
1890	1	2	0	2	0	1	1	2	0	2	1	1	2	2	17
1891	0	0	0	1	1	1	0	1	1	0	3	3	1	0	12
1892	1	3	2	0	1	1	3	0	1	1	0	1	1	0	15
1893	0	1	0	0	0	1	0	2	0	0	1	3	0	0	8
1894	1	0	0	0	0	0	0	0	1	0	1	1	0	0	4
Total	16	16	12	12	8	11	17	12	7	13	15	25	24	8	196

Of course, Monte Carlo tests also provide complete freedom in the choice of test statistic and, for a two–way table, one might adopt $u(x) = 1/\pi(x)$, the generalization of Fisher's statistic for 2×2 tables; see, for example, Besag (1992). For a more taxing application, testing for symmetry and independence, $\theta_{ij} = \phi_i \, \phi_j$, in square contingency tables, see Guo and Thompson (1994), which uses a similar listing of the data to generate samples, and unpublished notes by Besag and Seheult (1983), which uses a clumsier method.

In higher dimensions, complete independence is merely one of a wide range of hierarchical (especially graphical) models that we might wish to test and, in most such cases, there are no known direct methods of generating samples from the corresponding π's. In such cases, we must resort to MCMC exact p–values; see Besag and Clifford (1989, 1991) and Section 4.2 below. This is also true in the following application but first we note that in some contexts it is possible to construct exact Monte Carlo or MCMC confidence intervals; see Bølviken and Skovlund (1996) and references therein.

2.2.3 The Rasch model

Again we consider an $r \times s$ contingency table but now with the restriction that entries x_{ij} are binary. For example, in educational testing, $x_{ij} = 0$ or 1 represents the correct (1) or incorrect (0) response of candidate i to item j. Such tables are typically very large and it is more convenient here to present an application in evolutionary biology.

Ex. Darwin's finches

	Island identifier																	
Species	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Total
1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	14
2	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	13
3	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	14
4	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1	10
5	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0	12
6	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	2
7	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	10
8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
9	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	10
10	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	11
11	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	6
12	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17
Total	4	4	11	10	10	8	9	10	8	9	3	10	4	7	9	3	3	

The preceding table is taken from Sanderson (2000). Entries correspond to the presence or absence of 13 species of finch on 17 Galapagos Islands. A question of ecological concern is whether the frequencies with which the pairs of species occur together on the islands is very different from what one might expect by chance alone.

The most common statistical formulation for binary tables is the Rasch (1960) model. This asserts that all responses are independent and that the odds of 1 to 0 in cell (i, j) are $\theta_{ij} : 1$, where $\theta_{ij} = \phi_i \, \psi_j$, as in (19), though the ϕ_i's and ψ_j's no longer form probability distributions. Then the data x for r species (or candidates) and s islands (or items) has probability

$$\prod_{i=1}^{r} \prod_{j=1}^{c} \frac{\theta_{ij}^{x_{ij}}}{1 + \theta_{ij}} \;=\; \frac{\prod_i \phi_i^{x_{i+}} \, \prod_j \psi_j^{x_{+j}}}{\prod_i \prod_j \, (1 + \phi_i \, \psi_j)} \tag{20}$$

and the row and column totals, typically x_{i+} and x_{+j}, are again sufficient statistics for the ϕ_i's and ψ_j's. If we condition on these totals, we eliminate the unknown parameters and, in this case, obtain a uniform distribution $\pi(x)$ on the space S of allowable tables. Thus, an exact p–value for assessing the Rasch model against data $x^{(1)}$, using any particular test statistic $u(x)$, is given by the proportion of tables for which $u(x) \geq u(x^{(1)})$. However, enumeration is notoriously difficult, even for small tables, and is not feasible for the large tables that occur in educational testing. Furthermore, there are no known methods of creating a random sample of such tables, so that simple Monte Carlo tests do not provide an alternative and, once again, the only available option is MCMC, as we discuss in detail in Section 4.2. Finally, here, we note that an $r \times s$ binary table can be interpreted as one layer of an ordinary $2 \times r \times s$ contingency table in which the layer totals are all 1's, which enforces the binary restriction. The test of the Rasch model then becomes one of no three–way interaction in a (sparse) three–way table. Bunea and Besag (2000) discuss MCMC tests for more general $2 \times r \times s$ contingency tables.

2.3 Importance sampling

We have seen how to apply the notion of Monte Carlo sampling to learn about an otherwise intractable fixed p.m.f. π, both in parametric Bayesian inference and in calculating non–parametric frequentist p–values. However, in estimating parameters θ by the method of maximum likelihood, one is faced by the ostensibly more daunting task of dealing with a whole family of distributions, indexed by θ itself. Similarly, in Bayesian sensitivity analysis, one needs to assess the effects of changes to the basic assumptions. This may involve posterior distributions that have different functional forms and yet are not far apart, so that one would like to sample simultaneously from a whole family of distributions. In either context, the relevance of Monte Carlo methods is less obvious. Fortunately, *importance sampling* can often bridge the gap, because it enables one to approximate $\mathrm{E}_{\pi^*} g$ for distributions π^* that are close to the baseline distribution π from which we have a random sample. We now describe how this works.

In parallel to $\pi(x) = h(x)/c > 0$ for $x \in S$ in (1), consider another p.m.f.,

$$\pi^*(x) \;=\; h^*(x)\,/\,c^* \;>\; 0, \qquad\qquad x \in S^*,$$

where h^* is known and crucially $S^* \subseteq S$. Suppose that we require $\mathrm{E}_{\,\pi^*}g$, for a specific g, but that our random sample $x^{(1)}, \ldots, x^{(m)}$ is from π rather than π^*. Nevertheless, we can write

$$\mathrm{E}_\pi \frac{gh^*}{h} \;=\; \sum_{x \in S} \frac{g(x)h^*(x)}{h(x)}\,\frac{h(x)}{c} \;=\; \frac{c^*}{c} \sum_{x \in S^*} g(x)\,\frac{h^*(x)}{c^*} \;=\; \frac{c^*}{c}\,\mathrm{E}_{\pi^*}g, \tag{21}$$

so that the right–hand side of (21) can be estimated from $x^{(1)}, \ldots, x^{(m)}$ by the average value of $g(x^{(t)})h^*(x^{(t)})/h(x^{(t)})$. Usually, c^*/c is unknown but, as a special case of (21),

$$\mathrm{E}_\pi(h^*/h) \;=\; c^*/c,$$

so that, as our eventual approximation to $\mathrm{E}_{\pi^*}g$, we can adopt the ratio estimate,

$$\widetilde{\mathrm{E}}_{\pi^*}g \;=\; \sum_{t=1}^{m} w(x^{(t)})\,g(x^{(t)}), \tag{22}$$

where

$$w(x^{(t)}) \;=\; \frac{h^*(x^{(t)})/h(x^{(t)})}{\sum_{t=1}^{m}\{h^*(x^{(t)})/h(x^{(t)})\}}.$$

Note that the $w(x^{(t)})$'s are independent of g and are well defined because $S^* \subseteq S$. The estimate (22) should be satisfactory if (5) is adequate for $\mathrm{E}_\pi g$ and there are no large weights among the $w(x^{(t)})$'s. In practice, the latter condition requires that h and h^* are not too far apart. There are modifications of the basic method described here that can extend its range (e.g. umbrella sampling).

We illustrate importance sampling below and describe how it can be used for maximum likelihood estimation in Section 2.4 but first we note two applications to Bayesian inference. The first is sensitivity analysis, in which $\pi(x) = \pi(x|y)$ is a baseline posterior distribution and $\pi^*(x) = \pi^*(x|y)$ is a modified version of π. The second is less obvious and involves *sequential* importance sampling in which observations on a process arrive as a time series and the idea is to update inferences as each new piece of information is received, without the need to run a whole new simulation; see, for example, Liu and Chen (1998), Doucet *et al.* (2001), Liu (2001) and MacCormick and McCormick (2002).

Ex. Self–avoiding random walks

One of the earliest uses of Monte Carlo methods was in modeling chain polymers by self–avoiding random walks (Hammersley and Morton, 1954; Rosenbluth and Rosenbluth, 1955). Imagine a two–dimensional square lattice or a three–dimensional cubic lattice with unit spacing between its sites. Then we define a self–avoiding random walk of length r to be a sequence of r *distinct* sites, labeled $i = 1, \dots, r$, in which site 1 is at the origin and sites i and $i+1$ are adjacent for $i = 1, \dots, r-1$. The idea is that the sequence of sites should represent a chain of molecules connected by bonds. Here we concentrate on the original formulation in which all self–avoiding walks x of given length r have equal probability, even if this does not seem entirely realistic for the formation of polymers! Thus, the target distribution $\pi^*(x)$ is uniform on $x \in S^*$, the space of all self–avoiding walks of length r, and we might be interested in $\mathrm{E}_{\pi^*} g$, where $g(x)$ measures the *span* of x by the Euclidean distance between its endpoints.

For convenience of exposition, we consider the two–dimensional case. A naive method of simulating from π^* is as follows. Starting from site 1 at the origin, choose site 2 at random from its four neighbors. Then, for each $i = 2, \dots, r-1$, generate the next site $i+1$ at random from the three neighbors of i after excluding $i-1$. If a site is revisited at any stage of the process, the chain is abandoned and started again from scratch. This is a valid procedure because it is equivalent to generating chains of length r without testing for revisits, in which case each outcome has probability $\frac{1}{4}(\frac{1}{3})^{r-2}$, and then discarding those chains that are not self–avoiding. However, the algorithm becomes extremely slow as r increases, even for $r = 100$, say.

As a modification of the above procedure, we might choose site $i+1$, for $i = 1, \dots, r-1$, at random from the neighbors of i that have not been visited previously. If there are no such neighbors, the chain is abandoned and restarted as before. Although this algorithm is very much faster, it does not sample from the uniform distribution. Specifically, let q_i denote the number of neighbors available at stage i in a successful chain x of length r, so that $q_1 = 4$ and otherwise $q_i = 1$, 2 or 3. Then x has probability

$$\pi(x) \ \propto \ \prod_{i=1}^{r-1} q_i^{-1}, \qquad\qquad x \in S,$$

where, of course, in this case $S = S^*$. Nevertheless, if we generate a sample $x^{(1)}, \dots, x^{(m)}$ from π, we can use the formula (22) to adjust for the bias, with

$$w(x^{(t)}) \ = \ q^{(t)} / \sum_{t=1}^{m} q^{(t)},$$

where $q^{(t)}$ is the product of the q_i's for the tth chain.

In particular, we reweight a sample of size 500000 from π to obtain the approximation 25.96 for the expected span of a self–avoiding random walk with 100 sites; the unweighted average is 19.07, the average number of attempts per walk is 4.4 and the maximum weight is about 0.003. In a random sample of 1000 from π^* itself using the naive algorithm, we obtain 25.47, with a standard error 0.30, and even this requires serious CPU time, with an average of about 75000 attempts per walk and a maximum exceeding 500000! We return to the example in Section 3.3.1 and devise a simple Metropolis algorithm that produces the estimate 26.03 with a standard error of 0.03.

2.4 Monte Carlo maximum likelihood estimation

Let $x^{(0)}$ denote an observation from a parametrized p.m.f.,

$$\pi(x;\theta) \;=\; h(x;\theta)/c(\theta), \qquad x \in S, \quad \theta \in \Theta,$$

where $c(\theta)$ is a normalizing constant,

$$c(\theta) \;=\; \sum_{x \in S} h(x;\theta).$$

Suppose the true value of the parameter θ is unknown and we require its maximum likelihood estimate,

$$\hat{\theta} \;=\; \arg\max_{\theta \in \Theta} \, \pi(x^{(0)};\theta).$$

We assume that h is quite manageable but that $c(\theta)$ and its derivatives cannot be calculated directly, even for particular values of θ.

Nevertheless, suppose we can generate a random sample from $\pi(x;\theta)$ for any given θ. Let $x^{(1)}, \ldots, x^{(m)}$ denote such a sample for $\theta = \breve{\theta}$, a current approximation to $\hat{\theta}$. Then, trivially, we can always write

$$\hat{\theta} \;=\; \arg\max_{\theta \in \Theta} \, \ln \frac{\pi(x^{(0)};\theta)}{\pi(x^{(0)};\breve{\theta})} \;=\; \arg\max_{\theta \in \Theta} \left\{ \ln \frac{h(x^{(0)};\theta)}{h(x^{(0)};\breve{\theta})} - \ln \frac{c(\theta)}{c(\breve{\theta})} \right\}. \tag{23}$$

The first quotient on the right–hand side of (23) is known and the second can be approximated using (21), where $c(\theta)$, $c(\breve{\theta})$, $h(x^{(0)};\theta)$ and $h(x^{(0)};\breve{\theta})$ play the roles of c^*, c, h^* and h, respectively. That is,

$$\frac{c(\theta)}{c(\breve{\theta})} \;=\; \sum_{x \in S} \frac{h(x;\theta)}{c(\breve{\theta})} \;=\; \sum_{x \in S} \frac{h(x;\theta)}{h(x;\breve{\theta})} \, \pi(x;\breve{\theta})$$

can be approximated by the empirical average,

$$\frac{1}{m} \sum_{t=1}^{m} \frac{h(x^{(t)};\theta)}{h(x^{(t)};\breve{\theta})},$$

for any θ in the neighborhood of $\breve{\theta}$. It follows that, at least when θ is one– or two–dimensional, an improved approximation to $\hat{\theta}$ can be found by direct search, though, in higher dimensions, it is necessary to implement a more sophisticated approach, usually involving derivatives and corresponding approximations. In practice, several stages of Monte Carlo sampling may be required to reach an acceptable approximation to $\hat{\theta}$.

Unfortunately, in most applications where standard maximum likelihood estimation is problematic, so too is the task of producing a random sample from π. The above approach must then be replaced by an MCMC version. Our example below cheats by using perfect MCMC (see Section 4.6) to generate the random samples!

2.4.1 Maximum likelihood for the Ising model

Let X denote an $r \times s$ rectangular array of binary r.v.'s. In any particular realization x, define u to be the number of 1's and v to be the number of like–valued direct adjacencies on the array. Suppose that X has p.m.f.,

$$\pi(x;\theta) \;=\; \frac{\exp(\alpha u + \beta v)}{c(\theta)}, \qquad x \in S = \{0,1\}^{rs}, \tag{24}$$

where $\theta = (\alpha, \beta) \in \mathcal{R}^2$. This defines a two–dimensional finite–lattice *Ising model*, in which $\beta > 0$ promotes patches of 0's or 1's and u and v are jointly sufficient statistics for α and β. The Ising model, including its variants on other regular lattices in two or more dimensions, is of fundamental interest in statistical physics, where it has been studied extensively since the 1920's and, by MCMC methods, since the 1950's. For further details and an interesting historical account of (Markov chain) Monte Carlo methods, see Newman and Barkema (1999). Finite–lattice Ising models also provide examples of pairwise–interaction Markov random fields and, in particular, of the autologistic distribution in spatial statistics (Besag, 1974). It is easily established that the conditional distribution of any particular r.v. in (24), given the values of all others, depends only on the values of r.v.'s directly adjacent to it. The Ising model has been used quite widely in Bayesian image analysis as a somewhat crude prior distribution for object (1) against background (0), though this practice is open to criticism if the goal is anything more demanding than simple restoration (Tjelmeland and Besag, 1998).

The normalizing constant $c(\theta)$ in (24), called the partition function in statistical physics, cannot be evaluated by standard analytical or computational methods, unless the array is very small, except that it can be closely approximated on large arrays when $\alpha = 0$; that is, when the roles of the 1's and 0's are exchangeable. This is the case of most interest to physicists, because even moderately large values of β then induce substantial dependence between variables that are arbitrarily far apart. Indeed, on the infinite d–dimensional cubic lattice, with $d \geq 2$, there exists a critical value, $\beta^* = \ln(1+\sqrt{d})$, at and beyond which infinite patches of 0's or 1's occur, in apparent defiance of the conditional probability structure noted above. This phenomenon is called *phase transition* and its existence leads statistical physicists to use the Ising model to mimic spontaneous magnetization of a ferromagnet.

We now consider the maximum likelihood estimate of (α, β) in (24), based on a single realization $x^{(0)}$, giving values $u^{(0)}$ and $v^{(0)}$ of u and v. Then

$$h(x;\theta) \;=\; \exp(\alpha u + \beta v)$$

and (23) implies that $\hat{\theta}$ maximizes

$$(\alpha - \breve{\alpha})u^{(0)} + (\beta - \breve{\beta})v^{(0)} - \ln\{c(\theta)/c(\breve{\theta})\}, \tag{25}$$

where breves identify current approximations to the parameters. The Monte Carlo method enables us to apply the approximation,

$$\frac{c(\theta)}{c(\breve{\theta})} \;\approx\; \frac{1}{m}\sum_{t=1}^{m} \exp\{(\alpha - \breve{\alpha})u^{(t)} + (\beta - \breve{\beta})v^{(t)}\} \tag{26}$$

if we can draw an adequate random sample $x^{(1)}, \ldots, x^{(t)}$ from $\pi(x; \breve{\theta})$. As stated already, we can achieve this indirectly by borrowing a perfect MCMC sampler from Section 4.6. We consider a numerical example below but first we note that (genuine!) MCMC maximum likelihood can be applied to much more complicated Markov random fields than (24) and is not restricted to pairwise interactions. For an example on a hexagonal array, involving more than 20 parameters, see Tjelmeland and Besag (1998). More generally, although MCMC for maximum likelihood estimation has had much less impact than for Bayesian inference, it already has an important role in areas such as mixed effects models and no doubt its range of applications will continue to expand.

Ex. Initial pattern of disease in endives

These data concern the spread of a disease over a 179×14 approximately square–spaced array of endive plants and were first analyzed in Besag (1978). At the time, scientific interest centered mostly on spatial–temporal development of the disease: this is also the focus in Besag (2003). However, here we merely consider the initial pattern, coding the 2306 healthy plants by 0's and the 200 affected plants by 1's. As in the original paper, we simplify the analysis by conditioning on the data at the boundary sites. All those years ago, it seemed quite reasonable to model the pattern of disease for the 2124 interior plants, conditional on the boundary, by an Ising model with parameters estimated by the maximum pseudolikelihood method in Besag (1975). Although the author was aware of the Metropolis algorithm and had used it for synthesizing (24), perfect sampling and MCMC maximum likelihood had not yet been invented; and nor had MCMC goodness–of–fit tests, to which we return in Section 4.2.

The values of the sufficient statistics, conditioning on the boundary, are $u^{(0)} = 188$ and $v^{(0)} = 3779$. At each successive stage, we generate a Monte Carlo sample for the model (24), conditioned by the boundary and at the current approximate value $\breve{\theta}$ of $\hat{\theta}$. The eventual sample size is $m = 20000$ but smaller values of m are used earlier on. For each sample, we apply a Newton–Raphson algorithm in conjunction with (26) to obtain the next approximation to $\hat{\theta}$. Note that it may be necessary to do some recentering to avoid numerical problems. After several iterations, we obtain the estimates $\hat{\alpha} = -1.393$ and $\hat{\beta} = 0.299$, in reasonable agreement with the pseudolikelihood estimates -1.519 and 0.258.

The approximate standard errors are 0.240 and 0.078, obtained from the Fisher information matrix. We can also extend (24) to include an additional term γw, where w is the number of like–valued diagonal adjacencies, which is observed to be $w^{(0)} = 3940$. This leads to the estimates, $\hat{\alpha} = -0.973$, $\hat{\beta} = 0.254$ and $\hat{\gamma} = 0.175$, with approximate standard errors 0.292, 0.075 and 0.085. For comparison, the pseudolikelihood parameter estimates are -1.074, 0.233 and 0.163. Note that neither the numbers of decimal places nor the use of such large values of m are really warranted by the application; and also that genuine MCMC maximum likelihood would be more efficient than the pure Monte Carlo version presented here!

2.5 Simulated annealing

Let $\{h(x) : x \in S\}$, where S is finite, denote a bounded non–negative function, specified at least up to scale, and suppose our task is to find the "optimal" value $x^+ = \arg\max_x h(x)$. We assume for the moment that x^+ is unique but that S is too complicated for x^+ to be found by complete enumeration and that h does not have a sufficiently nice structure for x^+ to be determined by simple hill–climbing methods. In operations research, where such problems abound, h is often amenable to mathematical programming techniques; for example, the simplex method applied to the traveling salesman problem. However, here we make no such assumption.

Let $\{\pi(x) : x \in S\}$ denote the corresponding finite probability distribution defined by (1) and (2); in practice, c is usually unknown. Clearly, $x^+ = \arg\max_x \pi(x)$ and, indeed, the original task may have been to locate the global mode of π, as in our example below. Thus, our goal now is not to produce a random draw from π but to bias the selection overwhelmingly in favour of the most probable value x^+. The intention in simulated annealing is to bridge the gap between these two tasks.

The link is made by defining a corresponding *sequence* of distributions $\{\pi_k(x) : k = 1, 2, \ldots\}$, where

$$\pi_k(x) \;\propto\; \{h(x)\}^{m_k}, \qquad x \in S, \tag{27}$$

for an increasing sequence of m_k's. Then, each distribution has its mode at x^+ and, as k increases, the mode becomes more and more exaggerated. Thus, if we take a random draw from each successive distribution, there is increasing probability of producing the target x^+. Note the crucial point that this statement is unaffected by the existence of local maxima. If there are multiple global maxima, then eventually observations are drawn uniformly from among the x^+'s. Indeed, it was this fact that first suggested the existence of a second global maximum in the toy example with 20 components in Section 2.1.1!

For a more taxing illustration, we return to the second example in Section 2.1.1 with the same known values of α and β and the record $y = 1110011100\ldots$ of length 100000. We know already, via the Viterbi algorithm or otherwise, that the mode of $\pi(x|y)$ is at y itself but we now seek to show this via sampling from $\pi_k(x) \propto \{\pi(x|y)\}^k$. It is trivial to amend the original sampling algorithm to make draws from this distribution, though there are numerical complications if k becomes too large. Recall that, our single observation from $\pi(x|y)$ contained 22290 discrepancies with y. We now successively generate draws from $\pi_k(x)$ for $m_k = 2, 3, \ldots, 25$ and note the number of mismatches with y in each case. Thus, for $m_k = 2, 3, 4, 8, 12, 16, 20, 21, 22, 23, 24, 25$, we find 11928, 6791, 3826, 442, 30, 14, 0, 0, 2, 0, 0, 0 discrepancies, respectively. Although still a toy example, note that $\pi(y|y) \approx 5 \times 10^{-324}$, so the task is not entirely trivial from a sampling perspective.

Of course, in the real world, it is typical that, when x^+ cannot be found directly, nor can we generate draws from $\pi_k(x)$. In that case, one must apply an MCMC version of the above procedure, as described in Section 4.2.

3 Markov chain Monte Carlo calculations

3.1 Markov chains, stationary distributions and ergodicity

In ordinary Monte Carlo calculations, we are required to draw a perfect random sample from the target distribution $\{\pi(x) : x \in S\}$. We now assume that this is impracticable but that instead we can construct an ergodic (i.e. regular in the finite case) Markov transition probability matrix (t.p.m.) P with state space S and limiting distribution π and that we can obtain a partial realization from the corresponding Markov chain. Below we discuss some general issues in the construction and use of suitable t.p.m.'s but later we shall be much more specific, particularly in describing Hastings algorithms, of which Gibbs and Metropolis are special cases.

Thus, let $X^{(0)}, X^{(1)}, \ldots$ be a Markov chain with t.p.m. P and state space S and define $p^{(0)}$ to be the row vector representing the distribution of the initial state $X^{(0)}$. Then recall that the marginal distribution of $X^{(t)}$ is given by

$$p^{(t)} \;=\; p^{(0)} P^t, \qquad t = 0, 1, \ldots, \tag{28}$$

and that, if π is a probability vector satisfying *general balance* $\pi P = \pi$, then π is called a *stationary distribution* for P. That is, P maintains π and, if $p^{(0)} = \pi$, then $p^{(t)} = \pi$ for all $t = 1, 2, \ldots$. If, in addition, P is ergodic (i.e. irreducible and aperiodic), then π is unique and $p^{(t)} \to \pi$ as $t \to \infty$, irrespective of $p^{(0)}$. It then follows that $\bar{g}$, defined in (5) or, more correctly, the corresponding sequence of random variables, still converges almost surely to $\mathrm{E}_\pi g$ as $m \to \infty$, by the ergodic theorem for Markov chains. Furthermore, the sampling variance of $\bar{g}$ is of order $1/m$, though the estimate (6) is no longer valid because of the dependence. The underlying theory is more complicated than in ordinary Monte Carlo calculations but we can continue to use empirical averages to produce accurate approximations to expectations under π for sufficiently large m and we can quantify their precision.

In practice, stationarity, irreducibility and aperiodicity are somewhat separate issues in MCMC. Usually, one uses the Hastings recipe to identify a collection of t.p.m.'s P_k, each of which maintains π and is simple to apply but is not individually irreducible with respect to S. One then combines the P_k's appropriately to achieve irreducibility. In particular, note that, if $P_1, \ldots, P_n$ maintain π, then so do

$$P \;=\; P_1 P_2 \ldots P_n, \tag{29}$$

equivalent to applying $P_1, \ldots, P_n$ in turn, and

$$P \;=\; \frac{1}{n}\,(P_1 + \ldots + P_n), \tag{30}$$

equivalent to choosing one of the P_k's at random. Amalgamations such as (29) or (30) are very common in practice. For example, (30) ensures that, if a transition from x to x' is possible using a single P_k, then this is inherited by P. In applications of MCMC, where $x \in S$ has many individual components, $x_1, \ldots, x_n$ say, it is typical to specify a P_i for each i, where P_i allows change only in x_i. Then P in (29) allows change in each component in turn and (30) in any single component of x, so that, in either case, irreducibility is at least plausible.

Ideally, we would like to seed the chain by an $x^{(0)}$ drawn directly from π but of course, if we could do this, there would be no need for MCMC in the first place! Curiously, an exception to the general rule occurs in MCMC p–values, as we discuss later, but otherwise it is desirable to choose $x^{(0)}$ to be near the "center" of π. In any case, it is usual to ignore the output during a "burn–in" phase before collecting the sample $x^{(1)}, \ldots, x^{(m)}$ for use in (5). There are no hard and fast rules for determining the length of burn–in but assessment via formal analysis (e.g. autocorrelation times) and informal graphical methods, such as parallel box–and–whisker plots of the output, are usually adequate, though simple time–series plots can be misleading. This is an area of active research, including more theoretical approaches, such as Diaconis and Stroock (1991), Diaconis and Saloff–Coste (1993) and Roberts and Tweedie (1996).

There are some contexts in which burn–in is a crucial issue; for example, with the Ising model in statistical physics and in some applications in genetics. It is then desirable to construct special purpose algorithms; see, among others, Sokal (1989), Marinari and Parisi (1992), Besag and Green (1993), Geyer and Thompson (1995), Propp and Wilson (1996) and Murdoch and Green (1998). Some keywords include *auxiliary variables*, *multigrid methods*, *simulated tempering* (which is related to but different from simulated annealing), and *perfect simulation.* We return to some of these, with further references, in Section 4.

When X is high–dimensional, storage of MCMC samples can become a problem. Of course, (5) can always be calculated on the fly, for any given g, in which case no significant storage is required. However, in Bayesian applications, it is unusual for all g's of eventual interest to be foreseen in advance of the simulation. Because successive states $X^{(t)}$, $X^{(t+1)}$ usually have high positive autocorrelation, little is lost by *subsampling* the output. However, this has no intrinsic merit, contrary to some suggestions in the literature, and it is not generally intended that the gaps should be large enough to produce in effect a random sample from π. No new theory is required for subsampling: if the gap length is r, then P is merely replaced by the new Markov t.p.m. P^r. Therefore, we can ignore this aspect in constructing appropriate P's, even though eventually $x^{(1)}, \ldots, x^{(m)}$ in (5) may refer to a subsample stored after burn–in. Note also that burn–in and collection time are somewhat separate issues: the rate of convergence to π is enhanced if the second–largest eigenvalue of P is small *in modulus*, whereas a *large negative* eigenvalue can improve the efficiency of estimation. Indeed, one might use different samplers during the burn in and collection phases. See, for example, Besag *et al.* (1995), especially the rejoinder, for some additional remarks and references.

3.2 Detailed balance

We need to construct P's that satisfy general balance $\pi P = \pi$ with respect to π. That is, if $P(x, x')$ denotes the probability of a transition from $x \in S$ to $x' \in S$ under P, we require that

$$\sum_{x \in S} \pi(x)\, P(x, x') \;=\; \pi(x'), \tag{31}$$

for all $x' \in S$. However, there is an enormous advantage if we can avoid the generally intractable summation over the state space S. We can achieve this goal by demanding a much more stringent condition than (31), namely *detailed balance*,

$$\pi(x)\, P(x, x') \;=\; \pi(x')\, P(x', x), \tag{32}$$

for all $x, x' \in S$. Summing both sides of (32) over $x \in S$, detailed balance immediately implies general balance but the conditions (32) are much simpler to check, particularly if we stipulate that $P(x, x') = 0 = P(x', x)$ for the vast majority of $x, x' \in S$! Also note the trivial fact that (32) need only be checked for $x' \neq x$, which is important in practice because the diagonal elements of P are often complicated. The physical significance of (32) is that, if a stationary Markov chain $\ldots, X^{(-1)}, X^{(0)}, X^{(1)}, \ldots$ satisfies detailed balance, then it is *time reversible*, which means that it is impossible to tell whether a film of a sample path is being shown forwards or backwards. Incidentally, for theoretical investigations, it is sometimes helpful to rewrite (32) as a matrix equation,

$$\Delta P \;=\; P^T \Delta,$$

where Δ is the diagonal matrix with (x, x) element $\pi(x)$.

It is clear that, if $P_1, \ldots, P_n$ individually satisfy detailed balance with respect to π, then so does P in (30). Although time reversibility is not inherited in the same way by P in (29), it can be resurrected by assembling the P_i's as a random rather than as a fixed permutation at each stage; that is, in the trivial case $n = 3$,

$$P \;=\; \tfrac{1}{6}\,(P_1P_2P_3 + P_1P_3P_2 + P_2P_1P_3 + P_2P_3P_1 + P_3P_1P_2 + P_3P_2P_1).$$

The maintenance of time reversibility can have some theoretical advantages (e.g. the central limit theorem of Kipnis and Varadhan, 1986, and the initial sequence estimators of Geyer, 1992) and is worthwhile in practice if it adds a negligible computational burden.

3.3 Hastings algorithm

In a seminal paper, Hastings (1970) provides a remarkably simple general construction for t.p.m.'s P to satisfy detailed balance (32) with respect to π. Thus, let R be *any* Markov t.p.m. having state space S and elements $R(x, x')$. Now define the off–diagonal elements of P by

$$P(x,x') \;=\; R(x,x')A(x,x'), \qquad\qquad x' \neq x \in S, \tag{33}$$

where $A(x, x') = 0$ if $R(x, x') = 0$ and otherwise

$$A(x,x') \;=\; \min\left\{1,\; \frac{\pi(x')\,R(x',x)}{\pi(x)\,R(x,x')}\right\}, \tag{34}$$

with $P(x, x)$ defined by subtraction to ensure that P has unit row sums, which is legitimate because $\sum_{x'\neq x} P(x,x') \leq \sum_{x'\neq x} R(x,x') \leq 1$. Then, to verify that detailed balance (32) is satisfied for $x' \neq x$, either $P(x, x') = 0 = P(x', x)$ and there is nothing to prove or else direct substitution of (33) produces

$$\min\left\{\pi(x)R(x,x')\,,\,\pi(x')R(x',x)\right\}$$

on both sides of the equation. Thus, π is a stationary distribution for P, despite the arbitrary choice of R, though note that we might as well have insisted that zeros in R occur symmetrically. Note also that P depends on π only through $h(x)$ in (1) and that the usually unknown and problematic normalizing constant c cancels out. Of course, that is not quite the end of the story: it is necessary to check that P is sufficiently rich to guarantee irreducibility and aperiodicity with respect to π but usually this is straightforward.

It is useful to make a trivial extension to the above specification, by allowing R to be a rectangular matrix with elements $R(x, x')$ for $x \in S$ and $x' \in S'$, where $S' \supseteq S$. Of course, proposals $x' \notin S$ are always rejected because $\pi(x') = 0$. It may seem a little bizarre to allow proposals that are not in S but sometimes it simplifies acceptance probabilities; see Sections 3.3.1 and 3.6, for example.

Operationally, Hastings algorithms proceed as follows. When in state x, a *proposal* x^* for the subsequent state x' is generated with probability $R(x, x^*)$. Then either $x' = x^*$, with the *acceptance probability* $A(x, x^*)$, or else $x' = x$ is retained as the next state of the chain. Note that (33) does not apply to the diagonal elements of P: two successive states x and x' can be the same either because x happens to be proposed as the new state or because some other state x^* is proposed but is not accepted. This is therefore different from ordinary rejection sampling, where proposals are made until there is an acceptance, which would not be valid here.

3.3.1 Metropolis algorithm

Metropolis method (Metropolis, Rosenbluth, Rosenbluth, Teller and Teller, 1953) is the original MCMC algorithm and was developed for simulating particle systems in statistical physics, such as the Ising model. It corresponds to a Hastings algorithm in which $R(x, x^*) = R(x^*, x)$ for all $x, x^* \in S$, so that the acceptance probabilities $A(x, x')$ are independent of R! As an important special case, illustrated below, suppose that π is uniform on S. Then Metropolis proposals $x^* \in S$ are always accepted, which is merely a restatement of the fact that any symmetric t.p.m. P has a uniform stationary distribution. In Section 3.6, we describe Metropolis algorithms that operate only on individual components of x but our example here is not restricted in this way.

Ex. Self–avoiding random walks

We introduced self–avoiding random walks in Section 2.3. On a two–dimensional array, a walk x of fixed length r consists of r *distinct* lattice sites, labeled $i = 1, \ldots, r$, where site 1 is at the origin and sites i and $i+1$ for $i = 1, \ldots, r-1$ are directly adjacent. It is assumed that all such walks have equal probability, so the target distribution π is uniform on the space S of all self–avoiding random walks of length r; note that here we have dropped the superscripts $*$ of Section 2.3.

We construct a simple Hastings algorithm for self–avoiding random walks as follows. Let $x \in S$ denote the current configuration. To generate the next state x', we first choose an integer k uniformly at random from $1, \ldots, r-1$. If $k = 1$, we obtain x' by spinning the entire configuration about the origin through 90, 180 or 270 degrees, each with probability $\frac{1}{3}$. If $k \neq 1$, we propose spinning the partial configuration formed by sites $k+1, \ldots, r$ about site k through one of two angles, chosen uniformly from those above after excluding the one that moves site $k+1$ to coincide with site $k-1$. If the proposed spin produces a valid configuration, we accept it as x', else we set $x' = x$.

To show that the implied t.p.m. P is symmetric, consider any two states x and $x' \in S$. Then $P(x, x') > 0 \Rightarrow P(x', x) > 0$ and both moves require choosing the same site k and then the relevant spin with probability $\frac{1}{3}$ if $k = 1$ or else $\frac{1}{2}$, so $P(x', x) = P(x, x')$. To verify that P is irreducible, consider any of the four states in which the sites form a straight line. Then any other state can be reached in $r-1$ steps or less by choosing a subset of the sites $1, \ldots, r-1$ sequentially and performing a particular allowable spin. Reversing the sequence leads back to the original straight line. Hence all states communicate and P is irreducible with respect to S. It is easily seen that P is aperiodic. Therefore P is regular, with limiting distribution equal to its (unique) stationary distribution π. If r is sufficiently small (e.g. $r \leq 100$), we can initiate the chain by the naive random sampling algorithm in Section 2.3, else we can start in any valid state and allow a burn–in period.

We applied the algorithm with a run length of one million to estimate the expected span (distance apart of sites 1 and r) for a self–avoiding random walk with 100 sites, initiated by a random draw. We obtained the value 26.03 with an MCMC standard error 0.03 calculated using the initial sequence estimators in Geyer (1992). The average span agrees satisfactorily with the previous approximations in Section 2.3 and the standard error is an order of magnitude better than that obtained from the computationally much more costly 1000 random samples. The acceptance rate was about 52%. We repeated the exercise for $r = 500$ but it was impracticable to initiate the chain with a random draw and instead we used the biased algorithm in Section 2.3, which still required more than 60000 attempts. We then allowed a burn–in of 200000 samples, followed by a phase of length a million that produced an average span of 86.98 with a standard error of 0.14. The acceptance rate was about 37%.

In the above Metropolis algorithm, proposals outside S occur (provided $r \geq 5$). A modification is to choose k and then select a spin at random from the available spins that give a valid new configuration. If no valid spin is available, then the old configuration is retained or, equivalently, can be thought of as a last–ditch proposal that is always accepted. Note that it is *not* allowable when no non–null moves are available to try again with a new value of k because this destroys the uniform distribution on $1, \ldots, r-1$ and therefore violates the symmetry condition on R. Also note that the modified algorithm requires additional computation and may therefore be counterproductive for a fixed ration of CPU time, despite the gain in efficiency per iteration. For $r = 100$ and a run length of a million, we obtained an average span of 26.07, with an increased acceptance rate of 77% and a reduction of 12% in standard error.

3.4 Componentwise Hastings algorithms

In implementing a Hastings algorithm, how should R be chosen? Although there is always scope for insight and ingenuity, it is useful to have some general strategies to fall back on. The usual recipe, as has already been mentioned, is to construct a whole family of P_k's that maintain π and to use them in sequence or at random to ensure overall irreducibility. Each P_k then requires its own R_k and hence A_k, and the former can be chosen so that proposals and decisions on their acceptance are always comparatively simple and fast to make.

We now openly acknowledge that X has many components and write $X = (X_1, \ldots, X_n)$. We assume that each X_i is univariate, though this is not necessary. Then, the most common approach is to devise an algorithm in which an R_i is assigned to each individual component X_i. That is, if x is the current state, then R_i proposes a replacement x_i^* for the ith component x_i but leaves the remainder x_{-i} of x unaltered. Note that we can also allow some continuous components, in which case the corresponding R_i's and P_i's become transition kernels rather than matrices, with elements that are conditional densities rather than probabilities. Although the underlying Markov chain theory must then be reworked in terms of general state spaces (e.g. Nummelin, 1984; Meyn and Tweedie, 1993), the modifications in the practical procedure are entirely straightforward. For convenience here, we continue to adopt finite state space terminology.

In componentwise Hastings algorithms, the acceptance probability $A_i(x, x^*)$ for x_i^* can be rewritten as

$$A_i(x, x^*) \;=\; \min\left\{1,\, \frac{\pi(x_i^*|x_{-i})R_i(x^*, x)}{\pi(x_i|x_{-i})R_i(x, x^*)}\right\}, \tag{35}$$

which identifies the crucial role played by the *full conditionals* $\pi(x_i|x_{-i})$. Note that these n univariate distributions comprise the basic building blocks for Markov random field formulations in spatial statistics (Besag, 1974), where formerly they were referred to as the *local characteristics* of X. This connection explains why the use of MCMC methods in statistics originates in spatial applications.

The identification of the full conditionals from a given $\pi(x)$ follows from the trivial but, at first sight, slightly strange–looking result,

$$\pi(x_i|x_{-i}) \;\propto\; \pi(x) \;\propto\; h(x), \tag{36}$$

where the normalizing constant involves only a one–dimensional summation (or integration) over x_i. In any case, even this cancels out in the ratio (35) and, usually, so do many other terms simply because likelihoods, priors and posteriors are typically formed from products and then only those factors in (36) that involve x_i itself need to be retained. Such cancellations imply enormous computational savings, though they are not required for the validity of Hastings algorithms.

We also note that (36) generalizes to

$$\pi(x_A|x_{-A}) \;\propto\; \pi(x) \;\propto\; h(x), \tag{37}$$

where A is any given subset of $\{1, \ldots, n\}$. For example, (13) is a rather special case, with $A = \{k+1, \ldots, n\}$ and $\pi(x|y)$ replacing π. The immediate availability of such results is typical, even in highly complex formulations. Below we provide one of the simplest examples.

Ex. Autologistic and related distributions

As we noted in Section 2.4.1, the autologistic distribution (Besag, 1974) is a pairwise–interaction Markov random field for dependent binary data and can be interpreted as a generalization of the finite–lattice Ising model (24) that does not necessarily impose homogeneity and, indeed, is not tied to a regular lattice. There are at least two equivalent ways to parameterize the model: here we define the random vector $X = (X_1, \ldots, X_n)$ to have an autologistic distribution if the probability of the outcome $x = (x_1, \ldots, x_n)$ is given by

$$\pi(x) \;\propto\; \exp\,(\sum_i \alpha_i x_i + \sum_{i<j} \beta_{ij} 1[x_i = x_j]), \qquad x \in S = \{0, 1\}^n, \tag{38}$$

where the indices i and j run from 1 to n and the β_{ij}'s control the dependence in the system. The simplification with respect to a saturated model is that there are no terms involving three or more components in (38). Note that, in *graphical modeling*, the autologistic model appears under other names: thus, Cox and Wermuth (1994) refer to it as a *quadratic exponential binary distribution* and Jordan, Ghahramani, Jaakkola and Saul (1998) call it a *Boltzmann distribution*, following Hinton and Sejnowski (1986).

In most applications, a further reduction in the number of parameters is brought about perhaps by linking the α_i's via a linear model and, of particular interest here, by allowing only a small proportion of the β_{ij}'s to take nonzero values. Thus, in the Ising model itself, $\beta_{ij} = \beta$ for each pair of directly adjacent lattice sites i and j but is otherwise zero; in the noisy binary channel (17), $\pi(x|y)$ replaces $\pi(x)$ with y fixed, $\beta_{ij} = \beta$ whenever $|i - j| = 1$ and $\beta_{ij} = 0$ otherwise; and, for familial studies in epidemiology, β_{ij} might be nonzero only if individuals i and j are in the same household.

Quite generally, it follows from (36) and (38) that the full conditional distribution for X_i is given by

$$\pi(x_i|x_{-i}) \;\propto\; \exp\,(\alpha_i x_i \,+\, \sum_{j\neq i} \beta_{ij} 1[x_i = x_j]), \qquad x_i = 0, 1, \tag{39}$$

where we define $\beta_{ij} = \beta_{ji}$ for any $j < i$. Thus, the full conditional of X_i depends only on those X_j's for which $\beta_{ij} \neq 0$. In the terminology used for Markov random fields, a (possibly conceptual) *site* i is associated with each r.v. X_i and sites i and j are referred to as *neighbors* if and only if $\beta_{ij} \neq 0$.

The noisy binary channel (17) provides a particular instance of (38), in which

$$\pi(x_i|x_{-i}, y) \;\propto\; \exp\{\alpha 1[x_i = y_i] \,+\, \beta(1[x_i = x_{i-1}] \,+\, 1[x_i = x_{i+1}])\}, \tag{40}$$

where $x_0 = x_{n+1} = -1$ to accommodate the end points $i = 1$ and $i = n$. Thus, interior sites i have two neighbors, $i-1$ and $i+1$, whereas sites 1 and n each have one neighbor. Of course, both here and more generally in (39), it is trivial to evaluate the conditional probabilities themselves, because there are only two possible outcomes, but again we emphasize that the normalizing constant is not required in the Hastings ratio, which can be important in more complicated examples. Indeed, in the particular case of (40), there exist immediate extensions to higher dimensions, with applications to *Bayesian image analysis* (e.g. Geman and Geman, 1984). Also, there is no requirement for the x_i's or y_i's to be binary, the degradation mechanism can be much more complicated and α and β need not be known. For applications to tomographic image reconstruction, see Geman and McClure (1986) and, more comprehensively, Weir (1997).

Below, we discuss the two most widely used componentwise algorithms but first we remark that occasionally the capabilities of MCMC are undersold, in that the convergence of the Markov chain is not merely to the marginals of $\pi(x)$ (or $\pi(x|y)$) but to its entire multivariate distribution. Corresponding functionals (3), whether of a single component X_i or involving many components, can be evaluated with the same ease from a single run. Of course, there are some practical limitations: for example, one cannot expect to approximate the probability of some very rare event with high relative precision, without a perhaps prohibitively long simulation.

3.5 Gibbs sampler

The algorithm now known as the Gibbs sampler dates back at least to Suomela (1976) in his Ph.D. thesis on Markov random fields at the University of Jyväskylä. It was rediscovered independently by Creutz (1979) in statistical physics (where it is known as the *heat bath* algorithm), by Ripley (1979), again in spatial statistics, and by Grenander (1983) and Geman and Geman (1984) in their seminal work on Bayesian image analysis at Brown University and the University of Massachusetts. The term “Gibbs sampler” is due to Geman and Geman (1984) and refers to the simulation of *Gibbs distributions* in statistical physics, which correspond to Markov random fields in spatial statistics, the equivalence being established by the Hammersley–Clifford theorem (e.g. Besag, 1974).

The Gibbs sampler can be interpreted as a componentwise Hastings algorithm in which proposals are made from the full conditionals themselves; that is,

$$R_i(x, x^*) \;=\; \pi(x_i^*|x_{-i}), \tag{41}$$

so that the quotient in (35) is identically one and proposals are always accepted. The n individual P_i's are then combined as in (29), resulting in a *systematic scan* of all n components, or as in (30), giving a *random scan* sampler, or otherwise. The term “scan” is derived from applications in image analysis. Systematic and random scan Gibbs samplers are necessarily aperiodic, since $R_i(x, x) > 0$ for any $x \in S$. They are irreducible under the *positivity condition* $S = S_1 \times \ldots \times S_n$, where S_i is the *minimal* sample space for X_i; recall that S itself was defined to be minimal. Positivity holds in most practical applications and can be relaxed somewhat to cater for some of the exceptions. To see its relevance, consider the trite example in which $X = (X_1, X_2)$ and $S = \{00, 11\}$, so that no movement is possible using a componentwise updating algorithm. On the other hand, if $S = \{00, 01, 11\}$, then positivity is violated but both the systematic and random scan Gibbs samplers are irreducible. Severe problems occur most frequently in constrained formulations, such as the contingency table and Rasch model examples encountered in the section on Monte Carlo p–values.

Although the maintenance of the target distribution by a Gibbs sampler is ensured by the general theory for Hastings algorithms, there is a more direct and intuitive justification. This formalizes the argument that, if X has distribution π and any of its components is replaced by one sampled from the corresponding full conditional induced by π, to produce a new vector X', then X' must also have distribution π. That is, if x' differs from x at most in its ith component, so that $x'_{-i} = x_{-i}$, then

$$\Pr(X' = x') \;=\; \sum_{x_i} \pi(x)\,\pi(x'_i|x_{-i}) \;=\; \pi(x'_i|x_{-i})\,\pi(x_{-i}) \;=\; \pi(x').$$

Ex. Autologistic distribution

For a simple illustration, we return to the autologistic distribution (38) in Section 3.4. Then, for example, a single cycle of the systematic scan Gibbs sampler addresses each component x_i

in turn and updates it according to its full conditional distribution (39). Note that updates take effect immediately and not merely at the end of each cycle, else the limiting distribution would be incorrect. In Section 3.7, we consider a Bayesian example of Gibbs sampling that is a prototype for a wide variety of applications.

3.6 Metropolis algorithm revisited

We have noted already that the Metropolis algorithm is a special case of Hastings algorithm. In practice, individual components are usually updated one at a time, though this is certainly not a requirement. Then R_i for component i is chosen to be a symmetric matrix, so that the acceptance probability (35) becomes

$$A_i(x, x^*) \;=\; \min\{1,\, \pi(x_i^*|x_{-i})/\pi(x_i|x_{-i})\}, \tag{42}$$

independent of R_i! For example, if X_i takes on only a small number of values, then R_i might select x_i^* uniformly from these, usually excluding the current value x_i. If X_i is continuous, then it is common to choose x_i^* according to a uniform or Gaussian or some other easily–sampled symmetric distribution, centered on x_i and with a scale factor determined on the basis of a few pilot runs to give acceptance rates in the range 20 to 60%, say. A little care is needed if X_i does not have unbounded support, so as to maintain symmetry near an endpoint; this is an example in which proposals outside the minimal state space S may be made. Alternatively, a Hastings correction can be applied.

The main aim of Metropolis algorithms is to make proposals that can be generated and accepted or rejected very fast. Note that consideration of π arises only in calculating the ratio of the full conditionals in (42) and that this is generally a much simpler and faster task than sampling from a full conditional distribution, unless the latter happens to have a very convenient form. Thus, the processing time per step is generally much less for Metropolis than for Gibbs; and writing a program from scratch is much easier.

For a simple illustration, we again choose the autologistic distribution (38). Then, when updating x_i, the obvious proposal is $x_i^* = 1 - x_i$, the opposite of x_i, as suggested above. This is trivially a Metropolis procedure, with acceptance probability given by (42). Moreover, because $A_i(x, x^*) > \pi(x_i^*|x_{-i})$, it follows that the Metropolis algorithm for the autologistic distribution is generally more mobile than the Gibbs sampler and hence the former is statistically more efficient. This argument can be rigorized (Peskun, 1973, and more generally, Liu, 1996) and provides one good reason why in the past physicists preferred Metropolis to Gibbs (under the name "heat bath algorithm") for the Ising model. However, in this context, both algorithms have been superseded by versions of the Swendsen–Wang algorithm; see Section 4.5.

3.7 Gibbs sampling versus other Hastings algorithms

The Gibbs sampler has considerable intuitive appeal and one might assume from its popularity in the statistical literature that it represents an ideal among componentwise Hastings algorithms. However, we have just seen that this is not the case for the autologistic distribution. Indeed, an advantage bestowed by the more general Hastings formulation over the Gibbs sampler is that one can use the current value x_i of a component to guide the choice of the proposal x_i^* and to improve mobility around the state space S. For some further discussion, see Besag *et al.* (1995, Section 2.3.4) and Liu (1996). Even when Gibbs is *statistically* more efficient, a simpler algorithm may be superior in practice if 10 or 20 times as many cycles can be executed in the same run time. That is, traditional measures of efficiency are not necessarily relevant in comparing MCMC algorithms. The Hastings framework also enables one to easily consider vector proposals, which may be desirable in a quest to move more freely around the state space and is required in constrained formulations; see Section 4.2, for example. Multivariate proposals are also an essential ingredient in the Langevin–Hastings algorithm (Besag, 1994); see Section 4.4.

Having said all this, there are very many applications where efficiency considerations are relatively unimportant and in which the componentwise Gibbs sampler provides an entirely adequate computational tool. Furthermore, even when the (continuous) full conditional distributions are not easy to sample from by a standard method, they are often log–concave, in which case *adaptive rejection sampling* (e.g. Gilks, 1992) can be used. And there are occasions on which multivariate Gibbs steps can be implemented for some of the components without a large computational overhead, as, for example, in Cholesky decomposition for Gaussian components. Finally, in cases where Gibbs sampling is attractive in principle but awkward to implement, as is often the case for continuous components, it may be possible to rigorously adjust a discrete histogram approximation via Hastings steps; see Tierney, 1994, and, for related ideas involving *random* proposal distributions, Besag *et al.* (1995, Appendix 1). We now consider two examples, one that is and one that is not well suited to Gibbs sampling.

Ex. Toy example in Bayesian curve fitting

Suppose that observations $y_1, \ldots, y_n$ are available on an unknown function $g(t)$ at an evenly spaced sequence of time points $t = (t_1, \ldots, t_n)$. The observations are subject to independent Gaussian errors having unknown precision (inverse of variance) λ_y. The task is to estimate the underlying signal $\psi = (\psi_1, \ldots, \psi_n)$, where $\psi_i = g(t_i)$. The likelihood for ψ, given the data y, is therefore

$$L(y|\psi, \lambda_y) \;\propto\; \lambda_y^{\frac{1}{2}n} \exp\{-\tfrac{1}{2}\lambda_y \sum_{i=1}^{n} (y_i - \psi_i)^2\},$$

except that, in practice, we rescale the data, so that the working y_i's have crude variance one. This is rather untidy but is explained below. We assume g is fairly smooth and accordingly (e.g. Besag *et al.*, 1995) adopt a *locally quadratic* Gaussian prior $\rho(\psi|\lambda_\psi)$ for ψ; that is,

$$\rho(\psi|\lambda_\psi) \;\propto\; \lambda_\psi^{\frac{1}{2}n} \exp\{-\tfrac{1}{2}\lambda_\psi \sum_{i=2}^{n-1} (\psi_{i-1} - 2\psi_i + \psi_{i+1})^2\}, \tag{43}$$

where λ_ψ is an unknown precision parameter. We choose proper but rather vague independent Gamma(a, b) and Gamma(c, d) distributions as the priors for λ_y and λ_ψ, with $a = c = 1$ and $b = d = 0.005$. This implies that each precision has an exponential distribution with mean 200 but we retain generality in the equations below.

In (43), second differences $\psi_{i-1} - 2\psi_i + \psi_{i+1}$ of the ψ_i's are represented by independent Gaussian r.v.'s, having precisions λ_ψ, so the prior absorbs linear trends. However, its prime objective can best be seen from its full conditional density $\rho(\psi_i|\psi_{-i}, \lambda_\psi)$ in which the exponent is

$$\begin{aligned} &-\tfrac{1}{2}\lambda_\psi\{(\psi_{i-2} - 2\psi_{i-1} + \psi_i)^2 \;+\; (\psi_{i-1} - 2\psi_i + \psi_{i+1})^2 \;+\; (\psi_i - 2\psi_{i+1} + \psi_{i+2})^2\} \\ &= \; -3\lambda_\psi\{\psi_i - \tfrac{2}{3}(\psi_{i-1} + \psi_{i+1}) + \tfrac{1}{6}(\psi_{i-2} + \psi_{i+2})\}^2 \;+\; \ldots, \qquad i = 3, \ldots, n-2, \end{aligned}$$

with appropriate modifications at the ends. The weights $\frac{2}{3}$ and $-\frac{1}{6}$ imply that the conditional mean of this Gaussian distribution is obtained via the least squares fit of a quadratic equation to $\psi_{i-2}, \psi_{i-1}, \psi_{i+1}$ and ψ_{i+2}. The fact that the quadratic is fitted locally, rather than globally, explains the name of the prior. Also, the initial transformation of the y_i's, together with the impropriety of ρ, implies that the analysis is equivariant to affine transformations of the data, which seems desirable.

The above formulation produces the posterior density

$$\pi(\psi, \lambda|y) \;\propto\; L(y|\psi, \lambda_y)\, \rho(\psi|\lambda_\psi)\, \lambda_y^{a-1} \mathrm{e}^{-b\lambda_y}\, \lambda_\psi^{c-1} \mathrm{e}^{-d\lambda_\psi}, \qquad \psi \in \mathcal{R}^n,\ \lambda_y > 0,\ \lambda_\psi > 0, \tag{44}$$

for the underlying signal and the two precisions, given the data. This distribution can be interpreted as a hidden second–order Markov chain (with a continuous state space) and there are several ways in which ψ can be updated as a single block. A convenient though not very efficient method is via Cholesky decomposition, provided n is modest. With this in mind, we define the $n \times n$ matrix W by

$$\psi^T W \psi \;\equiv\; \sum_{i=2}^{n-1} (\psi_{i-1} - 2\psi_i + \psi_{i+1})^2,$$

where vectors, both here and below, are interpreted as column vectors, and the matrix Q by

$$Q \;=\; \lambda_y I \;+\; \lambda_\psi W,$$

where I is the $n \times n$ identity matrix. The full conditional for ψ is then

$$\psi \,|\, \lambda, y \;\sim\; \mathrm{N}\,(\lambda_y Q^{-1} y,\, Q^{-1}), \tag{45}$$

and those for λ_y and λ_ψ are

$$\lambda_y \,|\, \psi, \lambda_\psi, y \;\sim\; \text{Gamma}\,\{a + \tfrac{1}{2}n,\; b + \tfrac{1}{2}(y-\psi)^T(y-\psi)\}, \tag{46}$$

$$\lambda_\psi \,|\, \psi, \lambda_y, y \;\sim\; \text{Gamma}\,\{c + \tfrac{1}{2}n,\; d + \tfrac{1}{2}\psi^T W \psi\}. \tag{47}$$

On each cycle of the Gibbs sampler, we update ψ, λ_y and λ_ψ by sampling from their full conditionals (45), (46) and (47), with the new values always taking immediate effect. Within each cycle, it is preferable to update in a random order because this ensures the overall time reversibility of the algorithm. However, note that λ_y and λ_ψ are conditionally independent given ψ (and y). They can therefore be addressed simultaneously (in effect) and then time reversibility is preserved merely by updating ψ and λ in random order.

Numerical example. We generated 100 observations y from a $\text{N}(g(t), 4)$ distribution, where

$$g(t) \;=\; (\alpha \,+\, \beta \cos \omega t)\, \text{e}^{-\gamma t}, \qquad t = 1, \ldots, 100, \tag{48}$$

with $\alpha = 20$, $\beta = 10$, $\gamma = 0.01$ and $\omega = 0.6$. A block Gibbs sampler was run for 120000 cycles, with the first 20000 used as burn in. Every 20th sample of ψ and λ was stored, so that 5000 samples were available for subsequent calculations. The comparison (i.e. parallel box–and–whisker) plots for each of the 100 ψ_i's and the two $\ln \lambda$'s, in 10 batches of 500, are very stable, suggesting adequate run length.

The locally quadratic formulation does a reasonable job of recovering the true curve. In the posterior distribution of the ψ_i's, there are 8 values less than estimated 5% point and 3 greater than the 95% point, so that 89 values out of 100 lie within the corresponding 90% *pointwise* credible intervals; and 74 lie within the 80% intervals. One can also construct *simultaneous* credible curves: all the true values lie within the 60% and one lies above and one lies below the 50% envelope.

Although this example is ideally suited to block Gibbs sampling, one could of course use single site updating, which is much faster cycle by cycle but much less efficient per cycle. It is also interesting to contrast the full Gibbs sampler with an algorithm that employs a block Metropolis update of ψ. For example, given the current state (ψ, λ), one might choose the proposal $\psi^* = \psi + z$, where z is a random sample of n observations from a $\text{N}(0, \kappa)$ distribution and κ is chosen to produce a suitable acceptance rate. Then the quotient in the Metropolis acceptance probability becomes

$$\pi(\psi^*, \lambda | y) \,/\, \pi(\psi, \lambda | y) \;=\; \exp\{h(\psi) - h(\psi^*)\},$$

where

$$h(\psi) \;=\; \tfrac{1}{2}\lambda_y \sum_{i=1}^{n} (y_i - \psi_i)^2 \,+\, \tfrac{1}{2}\lambda_\psi \sum_{i=2}^{n-1} (\psi_{i-1} - 2\psi_i + \psi_{i+1})^2$$

which can be evaluated very fast.

Ex. Numerical example revisited. We repeated the analysis of the numerical example using the above Gibbs–Metropolis algorithm. A run of length 6 million dropping the first million and saving every 1000th sample, gave about 10% less accuracy but required rather less CPU time than the Gibbs sampler. The value of $\sqrt{\kappa}$ was 0.03, giving an acceptance rate of 30%. The timing issue is largely a reflection of the inefficiency of a very simple Gibbs program but note that the Metropolis version requires trivial changes to accommodate e.g. a non–Gaussian prior for ψ, whereas Gibbs sampling would usually demand a major effort. We also ran the Gibbs–Metropolis algorithm for an example with 500 rather than 100 observations. Again the results for a run length of 6 million were quite satisfactory. For example, 48 values out of 500 lie outside the 90% and 96 outside the 80% pointwise intervals.

Ex. Bayesian inference for the poly–Weibull distribution

In contrast to the usual applications of MCMC in Bayesian inference, the example below contains very few parameters and yet illustrates the problems that can occur in restricting MCMC to Gibbs sampling. It is prompted by a paper on competing risks models by Davison and Louzada–Neto (2000), which strongly criticizes the use of MCMC and, in particular, the Gibbs sampler, when more traditional approximations to posterior distributions are available. However, the paper is flawed, both in the Bayesian data analysis (though it provides a useful discussion of maximum likelihood) and in its failure to consider very simple MCMC alternatives to Gibbs. We use an example from Davison and Louzada–Neto's paper several times in Section 4 and begin here with some general background to basic problems in systems reliability and survival analysis.

Statistical models for the lifetime of a system (or of an individual) are often addressed in terms of the *hazard function* $h(.)$, where $h(t)$ is defined to be the instantaneous failure rate at time t, given survival at least to t. If we let

$$H(t) \;=\; \int_0^t h(u)\,du,$$

then it is easily shown that the survivor function $\bar{F}$ (the complement of the cumulative distribution function) and the probability density function f of lifetime are given by,

$$\bar{F}(t) \;=\; \mathrm{e}^{-H(t)}, \qquad f(t) \;=\; h(t)\,\mathrm{e}^{-H(t)}, \qquad t > 0. \tag{49}$$

Among simple models, one of the most common is the Weibull distribution, with

$$H(t) \;=\; (t/\theta)^{\beta}, \tag{50}$$

where $\theta > 0$ and $\beta > 0$ are scale and shape parameters. When a basic formulation is no longer adequate, a possibly appealing alternative is a *competing risks* model in which the system fails on expiry of the first of k independent (actual or conceptual) subsystems with individual simple hazard functions $h_1, \ldots, h_k$. In Section 4.7, we allow k to vary but here we take k as known, so that, in an obvious notation,

$$h(t) \;=\; \sum_{r=1}^{k} h_r(t), \qquad H(t) \;=\; \sum_{r=1}^{k} H_r(t), \qquad \bar{F}(t) \;=\; \prod_{r=1}^{k} \bar{F}_r(t). \tag{51}$$

As regards statistical inference in a competing risks model, we suppose that the h_r's are known in terms of a parameter vector $x = (x_1, \ldots, x_k)$, where each x_r may itself be a vector, and that independent observations $y_1, \ldots, y_n$ are available from n systems, though some of the y_i's are censored and do not represent actual failure times. For those systems in which failure does occur, it is not known which of the k subsystems has expired. Thus, with h and $\bar{F}$ given by (51), the likelihood function is

$$L(y, d \,|\, x) \;=\; \prod_{i=1}^{n} \bar{F}(y_i|x) \, \{h(y_i|x)\}^{d_i}, \tag{52}$$

where $d_i = 1$ if y_i is a failure time and $d_i = 0$ if y_i is a censored time. Our task is to make inferences about the properties of the underlying lifetime distribution from these partially censored data. In particular, Berger and Sun (1993) and Davison and Louzada–Neto (2000) discuss this for the *poly–Weibull* distribution, in which the expiry time for each subsystem r has a Weibull distribution with parameters θ_r and β_r. Note that, even for $k = 2$, the resulting four–parameter *bi–Weibull* distribution is sufficiently flexible to represent an interesting variety of qualitatively different hazard functions, including the celebrated "bathtub" curve in which $h(t)$ is initially decreasing, then goes through a relatively constant phase and is eventually increasing. This provides a substantial generalization of what can be achieved with the ordinary Weibull distribution.

We follow both Berger and Sun (1993) and Davison and Louzada–Neto (2000, Sections 3.2 and 4) in adopting the Bayesian paradigm, and especially the latter authors in fitting a censored bi–Weibull distribution to data in Lagakos and Louis (1988, Table 1) on the survival (sic) of 50 rats in a carcinogenesis experiment. However, here we implement a trivial Metropolis algorithm, rather than the cumbersome Gibbs sampler, which is described by Berger and Sun (1993), or Laplace's method, augmented by sampling–importance resampling, which is strongly advocated by Davison and Louzada–Neto (2000). Nevertheless, in Section 4.5, we return to Berger and Sun's paper, because it provides perhaps the earliest Bayesian example of an auxiliary variables reformulation in MCMC.

The likelihood function for the underlying poly–Weibull distribution is given by (52) with $H_r(t|x_r) = (t/\theta_r)^{\beta_r}$. For comparability with Davison and Louzada–Neto (2000), we adopt the same independent inverse exponential priors but other choices are equally straightforward and might be preferred. Because we are dealing with scale and shape parameters, it is natural to transform to

$$\phi_r \;=\; \ln \theta_r, \qquad \gamma_r \;=\; \ln \beta_r,$$

so that the prior for the $2k$–vector (ϕ, γ) becomes

$$\rho(\phi, \gamma) \;=\; \prod_{r=1}^{k} \{a_r \exp(-\phi_r - a_r \mathrm{e}^{-\phi_r} - \gamma_r - \mathrm{e}^{-\gamma_r})\}, \tag{53}$$

where the a_r's are specified constants. Then the posterior density $\pi(\phi, \gamma \,|\, y)$ of ϕ and γ, given the data y, is proportional to the product of (52) and (53), with the appropriate substitutions for h, H, θ and β in the expression for the likelihood. That is,

$$\pi(\phi, \gamma | y, d) \;\propto\; \prod_{i=1}^{n} \{\sum_{r=1}^{k} \beta_r y_i^{\beta_r - 1} / \theta_r^{\beta_r}\}^{d_i} \, \exp\{-\sum_{i=1}^{n} \sum_{r=1}^{k} (y_i/\theta_r)^{\beta_r}\}\, \rho(\phi, \gamma), \tag{54}$$

again with $\theta_r = \mathrm{e}^{\phi_r}$ and $\beta_r = \mathrm{e}^{\gamma_r}$. For Gibbs sampling, equation (54) is quite daunting, even when simplified by auxiliary variables: indeed, Berger and Sun (1993) additionally require log–concavity of the corresponding full conditionals. In contrast, it is trivial to program a Metropolis algorithm in which, at each successive stage, a proposal (ϕ^*, γ^*) is formed by adding $2k$ independent Gaussian variates, with mean zero and fixed variance σ^2, to the current (ϕ, γ) and accepting (ϕ^*, γ^*) as the next state with probability

$$\min\{1,\, \pi(\phi^*, \gamma^* | y, d) \,/\, \pi(\phi, \gamma | y, d)\},$$

else retaining (ϕ, γ). A Metropolis acceptance/rejection scheme arises because the proposal kernel, corresponding to the discrete t.p.m. R in (34), is symmetric. It is easy to choose a σ that produces an acceptance rate between about 20% and 60%. Note that the algorithm, which we refer to as *naive* Metropolis, does not require any derivatives or log–concavity in the prior or posterior or any awkward sampling. Of course, it is always possible to refine such a procedure. For example, mobility can be increased by assigning an individual σ to each component of (ϕ, γ) and sometimes it is preferable to propose updates of subsets of components or of single components. We comment later on further possible modifications and, in Sections 4.3.2 and 4.4.2, describe alternative Langevin–Hastings and auxiliary variables algorithms.

Davison and Louzada–Neto (2000) include three illustrative examples of the Laplace approximation (Tierney and Kadane, 1986) as an alternative to the Gibbs sampler used by Berger and Sun (1993). They claim that Laplace approximation "entails much less programming effort than does Markov chain Monte Carlo simulation, and there is no restriction to particular classes of priors". Both points are relevant to the Gibbs sampler but, as we have seen, the Metropolis method is trivial to program and does not require any particular properties of the prior. Davison and Louzada–Neto (2000) also comment "Laplace's method may fail if the posterior density is seriously multimodal, but we have not encountered this in the examples that we have tried".

Davison and Louzada–Neto's first two examples are simulations but the third involves real data on the lifetimes of 50 male rats (Lagakos and Louis, 1988, Table 1) exposed to 60 mg/kg of tuolene di–isocynate. The experiment was terminated after 108 weeks, with eight rats still surviving. The remaining 42 deaths occurred at times (in weeks) 2, 3, 5, 8, 8, 8, 9, 10, 12, 12, 14, 24, 24, 26, 38, 40, 42, 47, 52, 55, 60, 68, 70, 73, 74, 78, 79, 82, 82, 84, 90, 90, 90, 92, 96, 96, 100, 103, 103, 104, 105, 106. Cause of death was unknown but at least two possibilities were anticipated. Davison and Louzada–Neto (2000) fit a censored bi–Weibull distribution to the data, adopting the prior (53) with $a_1 = a_2 = 100$ for ϕ and γ. It follows that (ϕ_1, γ_1) is exchangeable with (ϕ_2, γ_2) in the posterior distribution $\pi(\phi, \gamma|y)$, though this point is overlooked in the paper.

A naive Metropolis algorithm, with Gaussian proposals and $\sigma = 0.2$, provides an acceptance rate of about 22%. Simple diagnostics show that $\pi(\gamma_1|y)$ and equivalently $\pi(\gamma_2|y)$ are severely bimodal, with correlation coefficient about -0.9 between γ_1 and γ_2. Figures 3(c) and 3(d) in Davison and Louzada–Neto (2000), which purport to show contour plots of the posterior densities of (ϕ_1, γ_1) and (ϕ_2, γ_2), respectively, are therefore incorrect and, at best, need to be amalgamated so as to represent either pair (ϕ_r, γ_r). The stated credible intervals are similarly defective. However, because the posterior modes for the γ_r are widely separated, the results in Davison and Louzada–Neto (2000) should correspond roughly with those for the ordered parameters defined in the next paragraph.

In fact, it is perhaps a little fortunate that a naive Metropolis algorithm succeeds in identifying the multimodality here. More often, one would expect such a simulation to become trapped for long periods in any pronounced mode of the target distribution, severely affecting performance. It is therefore important in general to identify potential problems, preferably before the simulation begins, and to tailor the MCMC algorithm accordingly. For example, when two subsets of the parameters are approximately exchangeable, one may additionally propose deterministic Metropolis swaps between their values on every cycle or every few cycles of the naive algorithm, as in Besag et al. (1995, Section 4). In the present setting of exact exchangeability, such proposals are always accepted, which can be counterproductive if, as usual, the output is subsampled. Among safer alternatives, one can instead propose a random reallocation of the subset values. Another possibility is to resolve the issue by imposing an ordering on the parameters. Thus, here with $k = 2$, one can redefine $\gamma_1 = \ln \min\{\beta_1, \beta_2\}$ and $\gamma_2 = \ln \max\{\beta_1, \beta_2\}$, which then requires that we modify the naive algorithm merely by additionally rejecting all proposals (ϕ^*, γ^*) for which $\gamma_1^* > \gamma_2^*$. There are obvious analogues of such devices that can be used more generally. However, a little care is needed. For example, it would not be valid here to generate (ϕ^*, γ^*)'s until $\gamma_1^* \leq \gamma_2^*$ and only then make the corresponding proposal, because this destroys the symmetry of R and therefore requires an awkward Hastings calculation. The use of uniform rather than Gaussian variates in creating proposals would simplify such a calculation but the computational overhead is probably not worthwhile.

The following results relate to the underlying bi–Weibull distribution, with the Davison and Louzada–Neto prior, applied to model the censored observations on the lifetimes of the 50 rats. Summaries from a single very long run for each of three different Metropolis algorithms are reported, corresponding to the naive (PW2MN), random reallocation (PW2MR) and ordered–parameter (PW2MO) versions described above. As noted already, the posterior distribution for the parameters using PW2MN and PW2MR is exchangeable between (θ_1, β_1) and (θ_2, β_2), with severe multimodality for the β's. There is little point in quoting estimates of the parameters but, for the record, the 95% equal–tailed credible intervals for θ_1, θ_2, β_1, β_2 are $(85.9, 367)$, $(85.8, 373)$, $(0.542, 10.6)$ and $(0.538, 10.4)$ under PW2MN and $(86.5, 368)$, $(86.1, 370)$, $(0.539, 10.4)$ and $(0.541, 10.4)$ under PW2MR. It seems that PW2MN swaps modes adequately without the need for random reallocation, though the very close agreement between the two outputs is perhaps spurious. For PW2MO, we can interpret (θ_1, β_1) and (θ_2, β_2) as representing separate components of risk. In the same order as before, the four posterior medians are 145, 109, 0.790, 5.44, which can be compared with the modal values 132, 109, 0.82, 6.8 given by Davison and Louzada–Neto (2000). The PW2MO 95% equal–tailed credible intervals are $(79.9, 461)$, $(96.8, 143)$, $(0.500, 1.19)$, $(2.05, 11.8)$, compared with Davison and Louzada–Neto's highest posterior density intervals $(84.1, 274)$, $(99.9, 123)$, $(0.57, 1.14)$ and $(3.71, 13.8)$. Note here that all our summaries are calculated from subsamples of 20,000 values collected at intervals of 500 with a burn–in of 2 million cycles. The vagueness of the data do not merit this amount of computing and we would usually store no more than 5,000 samples and quote 80 or 90% intervals rather than 95%. The PW2MO 90% intervals are $(86.3, 357)$, $(99.1, 129)$, $(0.539, 1.11)$, $(2.63, 10.3)$. We can also calculate 80% (say) simultaneous credible intervals for the parameters (Besag et al., 1995, Section 6.3) and here they are $(82.9, 410)$, $(97.9, 135)$, $(0.518, 1.15)$ and $(2.33, 11.1)$.

Perhaps of more interest, the table below provides approximations to the posterior mean of the probability of death occurring in each of the intervals, 0–2, 2–5, 5–10, 10–20, 20–30, . . ., 130–140, and > 140 weeks, based on Metropolis algorithms for the Davison and Louzada–Neto prior and three different likelihoods. Thus, the WeibM column employs the basic Weibull distribution, the next three columns fit bi–Weibull models, using naive, random reallocation and ordered Metropolis algorithms, respectively, and the PW3MO column refers to a three–component poly–Weibull formulation, with ordering. The final column is for a Langevin–Hastings algorithm and will be discussed in Section 4.3.2. Note that the bi–Weibull columns are estimating the same quantities and agree closely. The standard errors for the entries, calculated using the initial sequence estimators in Geyer (1992), are mostly around 0.0002, though some are a little larger. Of course, this is merely a statement about the accuracy of the MCMC and, in principle, arbitrarily small standard errors can be obtained using an appropriately long simulation. It is not surprising that the credible intervals for the probabilities are two orders of magnitude larger than the standard errors and again the very long runs adopted here have no practical merit.

The table shows clear discrepancy between the results for the Weibull and the bi–Weibull formulations but little between the bi–Weibull and the three–component poly–Weibull. As concluded by Davison and Louzada–Neto (2000), the fit of the bi–Weibull but not of the basic Weibull is in quite good agreement with the empirical distribution function for the observed data, up to censoring. Because the original data are limited to only 50 observations, it is instructive also to carry out a small simulation study, in which 500 censored observations are sampled from a bi–Weibull distribution with parameter values obtained from the data; that is, $\theta_1 = 145, \theta_2 = 109, \beta_1 = 0.790, \beta_2 = 5.44$. The resulting sets of 90% pointwise and 80% simultaneous credible intervals are $(114, 177)$, $(103, 109)$, $(0.755, 0.969)$, $(4.60, 6.76)$ and $(116, 171)$, $(103, 109)$, $(0.77, 0.95)$, $(4.70, 6.60)$, both just covering the correct values. The fitted lifetime distribution, corresponding to the table below, is also satisfactory. All 17 of the 90% pointwise credible intervals contain the true probabilities and indeed the same holds for the 70% (but not quite the 60%) simultaneous intervals. Incidentally, we changed the scale constant σ from 0.2 to 0.05 in analyzing the more informative simulated data.

Interval	WeibM	PW2MN	PW2MR	PW2MO	PW3MO	PW2LO
0 – 2	0.0140	0.0367	0.0367	0.0368	0.0370	0.0367
2 – 5	0.0248	0.0338	0.0337	0.0338	0.0337	0.0337
5 – 10	0.0449	0.0455	0.0454	0.0455	0.0456	0.0454
10 – 20	0.0939	0.0747	0.0746	0.0747	0.0753	0.0745
20 – 30	0.0927	0.0640	0.0639	0.0640	0.0651	0.0638
30 – 40	0.0878	0.0586	0.0585	0.0586	0.0600	0.0585
40 – 50	0.0811	0.0567	0.0567	0.0567	0.0581	0.0568
50 – 60	0.0737	0.0582	0.0582	0.0582	0.0594	0.0584
60 – 70	0.0661	0.0634	0.0635	0.0635	0.0640	0.0636
70 – 80	0.0586	0.0726	0.0727	0.0726	0.0722	0.0727
80 – 90	0.0516	0.0847	0.0849	0.0846	0.0834	0.0847
90 – 100	0.0450	0.0959	0.0961	0.0955	0.0943	0.0954
100 – 110	0.0390	0.0963	0.0962	0.0957	0.0958	0.0957
110 – 120	0.0336	0.0744	0.0742	0.0742	0.0747	0.0745
120 – 130	0.0289	0.0421	0.0422	0.0424	0.0416	0.0425
130 – 140	0.0247	0.0204	0.0206	0.0207	0.0199	0.0207
140 –	0.1395	0.0219	0.0219	0.0227	0.0200	0.0224

4 Some more specialized topics

In this section, we introduce some more specialized topics that are proving useful in statistical applications. The description of each is intended to be self–contained and can be read in isolation from the others. The topics are: adaptive slice sampling; MCMC p–values; MCMC maximum likelihood estimation (momentarily!); the Langevin–Hastings algorithm; auxiliary variables methods; perfect MCMC simulation; reversible jumps MCMC; and simulated annealing.

4.1 Adaptive slice sampling

In Section 4.4, we discuss *auxiliary variables*, first described in the statistical literature in general terms by Besag and Green (1993). The basic idea is to augment the original r.v. X, usually a vector, by additional ones to improve some aspect of the algorithm: perhaps simply the ease of writing the computer code or more ambitiously the ability of the algorithm to make large moves across the state space. Slice sampling (e.g. Higdon, 1994, 1998) is a special case and has some appealing theoretical properties; see Roberts and Rosenthal (1999) and, for a summary and further references, Mira and Roberts (2003). However, in its original form, it is awkward to implement. This problem is addressed by Neal (2003), with the introduction of *adaptive* slice samplers, which are far simpler computationally, to the extent that they can be used to construct generic MCMC algorithms that require very little tuning. Here, we first describe the slice sampler itself and then consider the simplest adaptive version.

Let X denote a single continuous r.v. with target p.d.f. $\pi(x) \propto h(x)$, where h is known, and define the auxiliary variable U, conditional on $X = x$, to have a uniform distribution on the interval $(0, h(x))$. Then X and U have joint p.d.f. proportional to $1[0 \leq u \leq h(x)]$ and the conditional p.d.f. of X, given $U = u$, is uniform on the possibly disjoint interval or *slice* defined by $\{x : h(x) \geq u\}$. It follows that, if x' is a draw from this conditional p.d.f. and we now repeat the process to generate a sequence $X, U, X', U', X'', \ldots$, then the r.v.'s $X, X', X'', \ldots$ all have marginal distribution π; indeed, we have described a Gibbs sampler between X and U. Furthermore, the univariate algorithm generalizes immediately to a multivariate target distribution, applying the same recipe to each of the corresponding full conditionals $\pi(x_i \,|\, x_{-i})$. The difficulty with this algorithm is that in general it is very costly to determine the successive slices, particularly if π has more than one mode.

We turn now to the adaptive slice samplers proposed in Neal (2003). Again we can concentrate on a single r.v. X with p.d.f. $\pi(x)$. As before, X is augmented by an additional r.v. but this has several components, which it is convenient to split into two parts U and V. Starting from the current x, the first component u_1 of the observed u is chosen uniformly from $(0, h(x))$ and therefore defines a slice just as in the ordinary sampler. However, there is no longer a need to identify the slice itself. Instead, either of two main procedures is used, each of which has two stages. Here we focus on the slightly simpler one, in which the stages are referred to as *stepping out* and *shrinkage*. The alternative version replaces the first stage by a *doubling* prcedure and then requires a more complicated shrinkage rule.

Recall that, in the most basic Metropolis algorithm, the proposal x^* is generated from x by sampling uniformly from an interval of fixed width, centered on x. In the new procedure, we again begin with an interval of fixed width w but its center u_2 is chosen as a draw from a uniform distribution on $(x - \frac{1}{2}w, x + \frac{1}{2}w)$; and, as we shall see, the choice of w is much less crucial. Then stepping out consists of adding intervals of length w to the left endpoint of the original interval until the current endpoint l satisfies $h(l) < u_1$; and similarly adding intervals to the right endpoint until the current endpoint r satisfies $h(r) < u_1$. Note that the interval (l, r) will not necessarily cover the entire u_1–slice unless h is unimodal. A useful modification is to specify a maximum number m of intervals that can be used in constructing (l, r), to draw u_3 uniformly from $\{0, 1, \ldots, m-1\}$ and to allow no more than u_3 add–ons to the left and $m - 1 - u_3$ to the right. The unrestricted case corresponds to $m = \infty$, whereas the opposite extreme is $m = 1$, in which case there are no add–ons and there is no adaptation. When using a finite m, we refer to the interval for $m = \infty$ as the *potential* (l, r) interval. Note that repeated use of uniform draws is critical in the eventual justification of the algorithm.

We now obtain the next value x'. The simplest procedure would be to draw values x^* uniformly from (l, r) until the first occasion that $h(x^*) \geq u_1$ and then use this value as x'. However, this can be very inefficient if the acceptance region is small in comparison to (l, r). Instead, Neal (2003) suggests shrinking the current interval after each unsuccessful draw. In detail, we first draw v_1 uniformly from (l, r) and set $x' = v_1$ if $h(v_1) \geq u_1$, else we shrink (l, r) to a new interval (v_1, r) if $v_1 < x$ or to (l, v_1) if $v_1 > x$; and we then repeat the procedure on the new interval, accepting $x' = v_2$ if $h(v_2) \geq u_1$, else shrinking the interval at its left or right end according to whether $v_2 < x$ or $v_2 > x$. Clearly the process must eventually terminate with an x'. If this occurs on trial k, so that $x' = v_k$, our eventual additional variable has components $u_1, u_2, u_3, v_1, \ldots, v_{k-1}$, deleting the redundant v_k, with u_3 also omitted if $m = \infty$. Note that, unlike the case in which (l, r) is used on each occasion, the shrinkage rule produces an x' that is *not* drawn uniformly from the intersection of (l, r) with the u_1–slice.

We check the validity of the above algorithm by verifying that detailed balance (32) is satisfied. First, for fixed w and m, note that knowledge of (x, u, v, x') enables us to reconstruct every detail of the path from x to x'. We now establish that there always exists a corresponding path from x' to x, with additional variable (u', v'); here u'_1 and u'_2 are drawn uniformly from $(0, h(x'))$ and from $(x' - \frac{1}{2}w, x' + \frac{1}{2}w)$, respectively, such that $u'_1 = u_1$ and the potential (l', r') interval coincides with the potential (l, r) seeded by u_2. When $m < \infty$, u'_3 is drawn uniformly from $\{0, 1, \ldots, m-1\}$ such that the *actual* stepping–out intervals coincide. It is easily seen from a simple diagram how this works and why it would not if the initial w–intervals were located symmetrically about x and x'.

To complete the reverse path from x' to x, we require that $v'_{<k} = v_{<k}$ and $v'_k = x$, so that the shrunken intervals always coincide and the path terminates at x. Note the crucial fact that, in obtaining x' from x, shrinkage cannot produce endpoints between x and x'. Thus, we have established a one–to–one correspondence between detailed paths from x to x' and from x' to x. Furthermore, in an obvious notation, $\pi(x)P(x, u, v, x') = \pi(x')P(x', u', v', x)$, because $\pi(x)$ and $\pi(x')$ are reduced to the same constant by the divisors $h(x)$ and $h(x')$ in the uniform densities for u_1 and u'_1. Hence, integrating over the additional variables, we have shown that detailed balance is satisfied with respect to π. As for the ordinary slice sampler, it is clear that the above univariate discussion extends immediately to a multivariate density $\pi(x)$.

Finally, note that irreducibility of the above sampler must be verified case by case. As a simple univariate example where it fails, suppose that $h(x) = 1$ for $1 < |x| < 2$ and $h(x) = 0$

otherwise. Then transitions between the separate parts of the distribution occur only if $w > 2$. This problem is alleviated if, instead of stepping out, the doubling procedure is used at the first stage.

Ex. Bayesian inference for the poly–Weibull distribution

We applied the adaptive slice sampler with stepping out to the poly–Weibull distribution described in Section 3.9. For both the two– and three–component versions, we used a burn in of 200000 and a collection phase of one million cycles. In each case, the Monte Carlo standard errors ware mostly 0.0001 and the agreement with the table in Section 3.9 is excellent, with occasional disparities in the fourth decimal place, as one would expect. The results here are more precise.

4.2 MCMC p–values

The notation here corresponds to that introduced in the earlier discussion of simple Monte Carlo p–values. Thus, our task is to determine whether an observation $x^{(1)}$ might reasonably have arisen from a distribution π. We now assume that we cannot sample directly from π but can construct a Markov t.p.m. P for which π is the limiting distribution. Note that there is an advantage over other MCMC applications in that we can use $x^{(1)}$ to seed the Markov chain. Then, if $x^{(1)}$ is indeed a draw from π, so are all subsequent observations, without any need for burn–in; that is, we are dealing with a stationary Markov chain. However, the problem we now encounter is that successive states are of course dependent and there is no obvious way in which to devise a legitimate p–value for the test. Note that the gaps required to produce effective independence may be prohibitive and, in any case, difficult to assess; furthermore, it is not the idea in MCMC to rely on independence.

Two remedies that retain an exact p–value, despite the dependence, are suggested by Besag and Clifford (1989); both involve running the chain *backwards*, as well as forwards, in time. Recall that a Markov chain is also Markov when time is reversed; that is, the distribution of the past, given the present and the future, depends only on the present. Furthermore, if the chain is stationary, the reversed chain has a transition probability matrix Q in which the probability of moving from $x \in S$ to $x' \in S$ is given by

$$Q(x, x') \;=\; \pi(x')\, P(x', x) \,/\, \pi(x)\,.$$

If P happens to be reversible, then (32) implies that $Q = P$ but this is not a necessary ingredient in either of the following devices, which we refer to as *parallel* and *serial* runs, respectively.

Suppose that, instead of running the chain forwards, we run it backwards from $x^{(1)}$ for r steps, using Q, to obtain a state $x^{(0)}$, say. Then we run the chain forwards from $x^{(0)}$ for r steps, using P, and do this $m-1$ times independently to obtain states $x^{(2)}, \ldots, x^{(m)}$ that are contemporaneous with $x^{(1)}$. It is clear that, if $x^{(1)}$ is a draw from π, then so are

$x^{(0)}, x^{(2)}, \dots, x^{(m)}$ but not only this: $x^{(1)}, \dots, x^{(m)}$ have an underlying joint distribution that is exchangeable, a property that must be inherited by the corresponding values $u^{(1)}, \dots, u^{(m)}$ of any particular test statistic $u = u(x)$. Thus, if $x^{(1)}$ is a draw from π, its rank among $u^{(1)}, \dots, u^{(m)}$ is once again uniform and can be used in the usual way. This procedure is rigorous because p–values are calculated on the basis of a correct model, which here implies that $x^{(1)}$ is from π. Note that $x^{(0)}$ must be ignored and that also it is not permissible to generate separate $x^{(0)}$'s, else $x^{(2)}, \dots, x^{(m)}$ are not exchangeable with $x^{(1)}$. The value of r should be large enough to provide ample scope for mobility around S, so that simulations can reach more probable parts of the state space when the model is incorrect. However, it is not essential for validity of the p–value that P be irreducible. Hence, the test can be used even when irreducibility is in question, as, for example, in some applications to multidimensional contingency tables.

For the serial version of the test, again suppose that $x^{(1)}$ is a draw from π. Now consider a chain with stationary distribution π, in which observations $y^{(1)}, \dots, y^{(m)}$ are taken at intervals of r steps, yielding values $u(y^{(1)}), \dots, u(y^{(m)})$ of the test statistic. Suppose we could arrange for $x^{(1)}$ to turn up in the dth position, so that $y^{(d)} = x^{(1)}$, where d is a draw from a uniform distribution on $1, \dots, m$. Then, marginally over d (but not conditionally), the rank of the observed test statistic $u^{(1)}$ among the $u(y^{(t)})$'s would be uniform and its observed rank would provide a legitimate p–value. This device can be implemented by first sampling d and then running the chain forwards from $y^{(d)} = x^{(1)}$ to obtain $y^{(d+1)}, \dots, y^{(m)}$ and backwards to obtain $y^{(d-1)}, \dots, y^{(1)}$. Note that there is no exchangeability in this version but that, in general, the serial test is more powerful than the corresponding parallel one with the same value of r, because almost all the samples are more steps away from $x^{(1)}$.

Finally, there are sequential versions of both tests. In the parallel case, there are no new considerations above those of simple sequential Monte Carlo tests. In the serial version, it is necessary, instead of choosing d, to arrange that at termination the position of the data $x^{(1)}$ among the available $y^{(t)}$'s is marginally uniform. This can be effected by at each stage either running forwards or backwards r steps from the current string of $y^{(t)}$'s to obtain the next member, the choice of direction being made according to easily prescribed probabilities.

In addition to the Rasch model, outlined previously, Besag and Clifford (1989) discuss two applications of MCMC p–values in spatial statistics. More recently, the approach has been applied in genetics (Guo and Thompson, 1994; Lazzeroni and Lange, 1997), in the analysis of square (Smith, Forster and McDonald, 1996) and multidimensional (Diaconis and Sturmfels, 1998; Bunea and Besag, 2000) contingency tables, in other forms of log–linear and logistic analyses (Forster, McDonald and Smith, 1996, 2003), and in tests for Markov chains (Besag and Mondal, 2004). However, some authors use MCMC as if it produces random samples and so their p–values are not strictly valid, though this could easily be rectified, as above. There is also occasional confusion between estimation of p–values and exact tests. Below we return to two previous datasets.

4.2.1 Exact p–values for the Rasch model

To construct a test for the Rasch model, we require an MCMC algorithm that maintains a uniform distribution on the space S of binary tables x with the same row and column totals as in the data $x^{(0)}$; and, also we would much prefer the algorithm to be irreducible with respect to S. Here we follow Besag and Clifford (1989). The simplest move that maintains the margins is depicted below, where a, $b = 0$ or 1.

$$\begin{array}{ccccc} & \vdots & & \vdots & \\ \cdots & a & \cdots & b & \cdots \\ & \vdots & & \vdots & \\ \cdots & b & \cdots & a & \cdots \\ & \vdots & & \vdots & \end{array} \qquad \rightarrow \qquad \begin{array}{ccccc} & \vdots & & \vdots & \\ \cdots & b & \cdots & a & \cdots \\ & \vdots & & \vdots & \\ \cdots & a & \cdots & b & \cdots \\ & \vdots & & \vdots & \end{array}$$

The two row indices and the two column indices are the same on the right as on the left. Of course, there is no change in the configuration unless $a \neq b$. It can be shown that any table in S can be reached from any other by a sequence of such switches, so that irreducibility is guaranteed. There are several possible ways in which the algorithm can proceed. The simplest is to choose two rows and two columns at random and to propose a swap if this is valid, else retain the current table. This defines a Metropolis algorithm and, since π is uniform, all proposals are accepted. A more sophisticated method is to keep a list of the rectangles in which non–null switches are valid. Let x denote the current table and suppose there are $r(x)$ available rectangles. Then we choose one of these at random, propose the corresponding move to table x^*, say, and accept x^* with the Hastings probability $\min\{1,\, r(x)/r(x^*)$, else retain x. It is easy to calculate the list of available switches when proposing x^* from those for x and hence find $r(x^*)$. Hence, the main additional overhead in this algorithm is the initial one of listing the possible switches from $x^{(0)}$; this could be substantial for a very large table.

Ex. Darwin's finches

We now return to the data in Section 2.2.3, taken from Sanderson (2000). This paper promotes a "Knight's Tour" algorithm and launches a scurrilous attack on statistics and those who sail in her; see also Sanderson, Moulton and Selfridge (1998), which includes the comparatively benign claim that "results from previous studies are generally flawed", including those based on the types of swaps described above, such as Manly (1995). However, as noted by Gotelli and Entsminger (2001), it is the Knight's Tour algorithm that is invalid because its limiting distribution is not uniform, producing meaningless p–values.

Of more interest here is the free choice of test statistic in analyzing datasets of this type. Gotelli and Entsminger (2001) provide an interesting discussion from an ecological perspective and advocate the use of a *co–occurrence index*. In particular, if $x_{ij} = 1$ or 0 according to whether species i is present or absent on island j, then the binary variable

$$a_{iji'j'} \;=\; x_{ij}(1-x_{i'j})\,(1-x_{ij'})\,x_{i'j'} \;=\; 1$$

if and only if i exists on j but i' does not, whereas i does not exist on j' but i' does. Then one possible co–occurrence index is the C–score statistic of Stone and Roberts (1990), proportional to $u(x) = \sum_{iji'j'} a_{iji'j'}$. We used the parallel version of Besag and Clifford's (1989) procedure with this statistic, allowing 10000 steps backwards, followed by 9999 runs of 10000 steps forwards. We found $u(x^{(0)})$ to be the largest of all 10000 values of the index, so that the null hypothesis is rejected at the 0.0001 level.

Note that, in its original context of educational assessment, one can devise some interesting test statistics for the Rasch model via the *coincidence matrix*, whose (j, j') element is the frequency with which candidates provide the same response to items j and j'. It is easy to define an appropriate $u(x)$ and our limited experience suggests that this can provide a powerful tool but note that the total score in the matrix is no use because it is fixed by the row and column totals of the data. One can also tailor statistics to test for differences between groups (e.g. by gender or race) or indeed modify the randomization to allow for differences between the groups. Such modifications would seem quite useful, especially in exploratory work, and also apply to ecological analyses.

Finally, we add some comments about irreducibility, though recall that this condition is not required for validity of MCMC exact p–values. Consider the trivial example below, in which we show the six 3×3 tables whose row and column totals are all unity:

$$\begin{array}{ccc}1&0&0\\0&1&0\\0&0&1\end{array}\qquad\begin{array}{ccc}1&0&0\\0&0&1\\0&1&0\end{array}\qquad\begin{array}{ccc}0&1&0\\1&0&0\\0&0&1\end{array}\qquad\begin{array}{ccc}0&0&1\\0&1&0\\1&0&0\end{array}\qquad\begin{array}{ccc}0&1&0\\0&0&1\\1&0&0\end{array}\qquad\begin{array}{ccc}0&0&1\\1&0&0\\0&1&0\end{array}$$

Clearly, these tables communicate by simple exchanges but now suppose that we restrict the elements on the leading diagonal to be zero, so that only the final two tables are legal, neither of which permits a valid exchange. Then, in order to retrieve irreducibility, both here and in the general case of an $r \times s$ table with structural zeros, we must either devise some new moves or equivalently retain the old ones, allow the algorithm to make excursions outside the target space S and subsequently delete all of the illegal tables from the output. Note that, in our example, we need only add the 4th table (say) to S and not all the others. Similar economies are crucial in the general case, else the chain can become lost interminably in the augmented states. It is shown in Bunea and Besag (2000) that it suffices to allow one structural zero at a time to take the illegal value one. It may also be necessary to introduce a Hastings correction to ensure that the limiting distribution under the restriction to S.

Ex. Conditional Ising model for the endives data

In Section 2.4.2, we fitted Ising models (24) to the endives data, estimating the parameters by Monte Carlo maximum likelihood. We now address the more basic question of whether, again when we condition on the observed boundary values B, the interior pattern of disease is consistent with such the two–parameter model. We eliminate α and β from (24) by also conditioning on the sufficient statistics u and v. This leaves us with a distribution $\{\pi(x|u,v,B) : x \in S\}$ that is uniform on a very complicated space S, consisting of binary arrays with the same boundary values, the same number of 1s and the same number of like–valued adjacencies as in the observed data. Thus, our main task in obtaining an MCMC p–value for the model is to construct a transition probability matrix whose stationary distribution is $\pi(.|u,v,B)$. We achieve this via a trivial Metropolis algorithm, in which, at each successive stage, we choose two interior sites at random and swap their values if this maintains v, else we leave the array unaltered. The corresponding transition probability matrix is therefore symmetric, which implies that it maintains the required uniform distribution. A simple modification of the algorithm is to propose a swap between a randomly selected zero and a randomly selected one at each stage. However, in general, these algorithms are not irreducible with respect to S. As a toy example, the two 4×4 arrays

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \qquad\qquad \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

have the same borders and the same values of u and v but do not communicate via simple swaps. Fortunately, as remarked previously, the validity of MCMC p–values does not require irreducibility with respect to the original state space S. Of course, it would be of interest to devise a practicable algorithm that does ensure irreducibility but in practice it seems that simple swaps permit a reasonable amount of mobility, toy examples apart!

For the endives data, we implemented the serial version of the test, with 999 simulated arrays at gaps of 10000 steps. The value of d was 452. As test statistic, we chose the number of like–valued diagonal adjacencies, which for the data is 3944. The simulations produced 13 values greater than this and a further 11 tied values, so there is substantial conflict between the data and the model, with a p–value between 0.014 and 0.025. This suggests the extension of (24) that includes a parameter γ for diagonal adjacencies. A corresponding run provided no evidence against the modified formulation. It is also possible to construct MCMC tests for data on the subsequent spread of the disease.

4.3 MCMC maximum likelihood estimation

We have little to say here apart from the obvious fact that the Monte Carlo samples of Section 2.4 are replaced by MCMC samples. For further discussion, see Geyer (1991) and Geyer and Thompson (1992). As we have stated already, the example in Section 2.4 was really MCMC, with many parallel runs. However, we did rerun the endives example, starting with a perfect draw and then running a single chain for one million iterations. Of course, the results were consistent with the previous ones but more precise because of the long run length and the limited dependence.

4.4 Langevin–Hastings algorithm

The Langevin–Hastings algorithm, introduced in Besag (1994), provides a rigorous method of simulating from a continuous multivariate distribution π using vector proposals. Thus, suppose that

$$\pi(x) \;\propto\; \exp\{-u(x)\}, \qquad x \in S = \mathcal{R}^n, \tag{55}$$

and that $\nabla u(x)$, the vector of partial derivatives of u, exists throughout R^n. Now consider the stochastic differential equation,

$$dx(t) \;=\; -\nabla u(x(t))\, dt + \sqrt{2}\, dw(t), \tag{56}$$

where t denotes continuous time and $w(t)$ is standard n–dimensional Brownian motion. This is a special case of the n–dimensional Fokker–Planck equation and defines *Langevin diffusion*. It is easily established that (56) has limiting distribution π in (55) and this has motivated the use of discrete–time MCMC approximations, in which the current state x is replaced by a new state,

$$x' \;=\; x - \tau\nabla u(x) + z\sqrt{2\tau}, \tag{57}$$

where τ is a small positive time constant and z is a random sample of size n from a standard Gaussian distribution; see, for example, Amit, Grenander and Piccioni (1991). However, the errors in the approximation may accumulate to produce a limiting distribution that is far removed from π. Fortunately, this problem can be easily rectified by using x' merely as a Hastings proposal x^* for the next state, which ensures that the stationary distribution for the modified sampler is exactly π. Note that this also provides considerable flexibility. For example, it is allowable to increase τ so as to make appreciable moves, so long as the acceptance probability (34) does not become too small, or indeed to assign a distribution to τ; also, proposals need not be Gaussian. For theoretical results on the convergence of Langevin and Langevin–Hastings algorithms, see Roberts and Tweedie (1996).

Note that the Langevin–Hastings algorithm is not directly applicable to a random vector X whose density is positive only on part of R^n. However, it may be possible to use the algorithm to simulate a transformed version of X and then back–transform the output. In particular, a componentwise logarithmic transformation may work if $\pi(x) > 0$ only for $x \in R_n^+$. For an application of the algorithm to spatial point processes, see Møller, Syversveen and Waagepetersen (1998).

Ex. Multivariate Gaussian distribution

As a purely illustrative example, we construct a Langevin–Hastings algorithm for a random vector X having an n–dimensional multivariate Gaussian distribution with mean μ and precision matrix Q. Then

$$u(x) \;=\; \tfrac{1}{2}(x-\mu)^T Q(x-\mu), \qquad\qquad \nabla u(x) \;=\; Q(x-\mu)$$

and (57) implies that the proposal from a current state x is

$$x^* \;=\; x - \tau Q(x-\mu) + z\sqrt{2\tau},$$

where z is a random sample from a N(0,1) distribution, say. Then x^* is accepted as the next state x' with probability (34), else $x' = x$. Of course, in practice, one would usually adopt an exact procedure for sampling from a Gaussian distribution, based on Cholesky decomposition, for example. Nevertheless, the Langevin–Hastings algorithm might be of interest in some applications to Gaussian Markov random fields, where Q is a large but sparse matrix.

Ex. Bayesian inference for the poly–Weibull distribution

In Section 3.7.1, we described Bayesian inference for data from a censored poly–Weibull distribution and the construction of corresponding Metropolis algorithms. Here we discuss implementation of a Langevin–Hastings algorithm. Quite generally, suppose that the underlying lifetime distribution has hazard function h and survivor function H, parametrized by x. Then, a random sample $y_1, \ldots, y_n$, with censoring at t_0, implies that, apart from a constant,

$$u(x) \;=\; -\sum_i \delta_i \ln h(y_i) + \sum_i H(y_i) - \ln\rho, \tag{58}$$

where δ_i is defined as in equation (52) and h, H and the prior ρ are functions of x. In the competing risks framework, h and H are given by (51), with each pair (h_r, H_r) depending on distinct sets of parameters. Then, if ψ_r denotes any particular parameter associated with subsystem r,

$$\frac{\partial u(x)}{\partial \psi_r} \;=\; -\sum_i \frac{\delta_i}{h(y_i)}\frac{\partial h_r(y_i)}{\partial \psi_r} - \sum_i \frac{\partial H_r(y_i)}{\partial \psi_r} - \frac{\partial \ln\rho}{\partial \psi_r} \tag{59}$$

and is a typical element of the gradient vector $\nabla u(x)$ in equation (57). For the poly–Weibull distribution, as formulated in Section 3.7.1, $H_r(t) = (t/\theta_r)^{\beta_r}$ and there are k separate pairs of parameters, with $\psi_r = \phi_r = \ln\theta_r$ or $\psi_r = \gamma_r = \ln\beta_r$. Then,

$$\frac{\partial h_r(y_i)}{\partial \phi_r} \;=\; -\beta_r h_r(y_i), \qquad \frac{\partial h_r(y_i)}{\partial \gamma_r} \;=\; h_r(y_i)\{1 + \beta_r \ln(y_i/\theta_r)\},$$

$$\frac{\partial H_r(y_i)}{\partial \phi_r} \;=\; -\beta_r H_r(y_i), \qquad \frac{\partial H_r(y_i)}{\partial \gamma_r} \;=\; \beta_r H_r(y_i)\ln(y_i/\theta_r),$$

and, if as before we adopt the Davison and Louzada–Neto (2000) prior,

$$\frac{\partial \ln \rho}{\partial \phi_r} \;=\; a_r \mathrm{e}^{-\phi_r} - 1, \qquad\qquad \frac{\partial \ln \rho}{\partial \gamma_r} \;=\; \mathrm{e}^{-\gamma_r} - 1.$$

It is now straightforward to construct the basic Langevin–Hastings algorithm for data from the censored distribution and also to implement modified algorithms, corresponding to those in Section 3.7.1.

In particular, we again analyze the data on the lifetimes of rats, taken from Lagakos and Louis (1988), and model these by the censored bi–Weibull distribution ($k = 2$). We refer to the three variants as PW2LN, PW2LR and PW2LO and use the same run lengths as before, though these demand rather more CPU time. The numerical results are very close to the previous ones, so that, for example, the 90% equal–tailed credible intervals for θ_1, θ_2, β_1, β_2 using PW2LO are $(86.2, 358)$, $(99.2, 129)$, $(0.538, 1.12)$, $(2.64, 10.3)$. Additionally, the final column of the table in Section 3.7.1 provides the posterior means for the probabilities of death in successive intervals, obtained from PW2LO. The standard errors are broadly in agreement with those for the Metropolis algorithms, so that, given the additional CPU time, the performance of the Langevin–Hastings algorithm is rather disappointing. The results for PW2LN and PW2LR are comparable with those for PW2LO, apart from two of the PW2LN estimated posterior means in the table, both flagged by larger standard errors, 0.0007 and 0.0010.

4.5 Auxiliary variables

As usual, let $\{\pi(x) : x \in S\}$ denote the probability distribution of a multicomponent random quantity X for which we require an MCMC sampler. Suppose that standard irreducible componentwise algorithms are unsatisfactory, because they do not move fast enough around S. For example, this occurs in the Ising model (24) if $\alpha = 0$ and β is close to the critical value β^*; it also occurs beyond β^* but a simple fix is then available. To combat slow mobility, it is desirable that a sampler incorporates simultaneous updates of large blocks of conditionally dependent components but, of course, simple grouping generally leads precisely to the problems that MCMC is intended to avoid.

A possible alternative is to introduce a vector of entirely conceptual *auxiliary* r.v.'s into the simulation procedure, with the aim of decoupling the complex dependencies that exist among the components of X. This may require much ingenuity and, as yet, there have been relatively few success stories, the most notable being the Swendsen and Wang (1987) algorithm for Ising and Potts models. Nevertheless, the basic idea shows considerable promise and is exploited in a Bayesian setting by Damien, Wakefield and Walker (1999).

A general description of auxiliary variables is as follows. Imagine that, given the current state x of X, we create a (discrete) random vector R, whose conditional distribution $\nu(r|x)$ is under our control. Then, given $X = x$ and $R = r$, we define the subsequent state $x' \in S$

to be drawn from the conditional distribution $\eta(x'|x,r)$, required to satisfy

$$\pi(x)\,\nu(r|x)\,\eta(x'|x,r) \;\equiv\; \pi(x')\,\nu(r|x')\,\eta(x|x',r). \tag{60}$$

It follows that time reversibility between X and X' is satisfied, since, if X has marginal distribution π, then

$$\begin{aligned}\Pr(X=x,\, X'=x') \;&=\; \sum_r \pi(x)\,\nu(r|x)\,\eta(x'|x,r)\\ &=\; \sum_r \pi(x')\,\nu(r|x')\,\eta(x|x',r) \;=\; \Pr(X=x',\, X'=x).\end{aligned}$$

We can now iterate the procedure to produce a sequence $X, R, X', R', X'', \ldots$ say, so that the subsequence $X, X', X'', \ldots$ forms a Markov chain with marginal distribution π. If the implied t.p.m. is ergodic, then π is its limiting distribution, regardless of the initial $X \in S$. Unfortunately, this does not provide a recipe for choosing ν and primarily one must proceed by example, though Edwards and Sokal (1988) suggest some general guidelines; see also Besag and Green (1993).

An important special case of auxiliary variables arises if η is chosen to be the conditional distribution of X, given R, induced by their joint distribution; that is,

$$\eta(x|x',r) \;\propto\; \pi(x)\,\nu(r|x).$$

Then clearly (60) is satisfied and indeed the algorithm is a *block* Gibbs sampler between X and R. Here, a sufficient condition for ergodicity is that there exists an r^* such that $\nu(r^*|x) > 0$ for all $x \in S$.

Before discussing the Swendsen–Wang algorithm in detail, we make a few remarks about generalizations of auxiliary variables. The first of these is to auxiliary *processes*, introduced by Geyer (1991). Consider a target distribution $\{\pi(x) \propto h(x) : x \in S\}$ and now define a corresponding family $\{\pi_k : k = 0, 1, \ldots, m\}$, where, for example,

$$\pi_k(x) \;\propto\; \{h(x)\}^{k/m}, \qquad x \in S, \tag{61}$$

so that, at one extreme, we have the target distribution $\pi = \pi_m$ and, at the other, a comparatively simple distribution π_0, here uniform, for which a componentwise sampler has adequate mobility. Suppose now that we run componentwise MCMC algorithms for all $m+1$ processes in parallel but also make occasional proposals to swap the current states of a randomly selected pair of adjacent chains. That is, if chains k and $k+1$ are chosen, then their current states, here referred to as x_k and x_{k+1}, are swapped with the Metropolis acceptance probability,

$$\min\left\{1,\, \frac{\pi_k(x_{k+1})\,\pi_{k+1}(x_k)}{\pi_k(x_k)\,\pi_{k+1}(x_{k+1})}\right\}, \tag{62}$$

or else left as they are. The intention is that the mobility of the sampler for π_0 should be inherited by the other chains, via the swaps, and, in particular, by the sampler for π. Note that the individual chains no longer have the Markov property; also that, of course, if π_0 can be sampled exactly, then this can be used to advantage.

A closely related notion is that of *simulated tempering*, due to Marinari and Parisi (1992); see also Geyer and Thompson (1995). Again, this involves a hierarchy of distributions, such as (61), but only a single chain is run, with its level k changing stochastically. Data are retained only when $k = m$, so that storage requirements are modest. Inference about π involves ratio estimators, for which the theory is very simple if π_0 can be sampled exactly, because entries into level 0 are *regeneration points*. A disadvantage of simulated tempering is that approximate information on the normalizing constants for the π_k's must be collected beforehand, whereas the constants cancel out in (62). Incidentally, we note that both ideas have loose connections with simulated annealing and reversible jumps.

Ex. Swendsen–Wang algorithm

The most successful application of auxiliary variables methods thus far is the Swendsen and Wang (1987) algorithm for the Ising model. Here we describe the trivial generalization to the autologistic distribution (38) with non–negative interactions β_{ij}. Thus, let $x \in S$ denote the current state of the system. Then we introduce a set of conditionally independent auxiliary r.v.'s $R_{ij}=0$ or 1 for $i < j$, satisfying

$$\Pr(R_{ij} = 1|x) \;=\; 1 - \exp(-\beta_{ij}\, 1[x_i = x_j]) \;=\; p_{ij},$$

say. It is here that we require that $\beta_{ij} \geq 0$ so that $0 \leq p_{ij} < 1$. If $R_{ij}=1$, we say that sites i and j are *bonded*. Note that this can occur only if i and j are neighbours in the Markov random field sense (i.e. $\beta_{ij} \neq 0$) and additionally $x_i = x_j$. The bonds partition the sites into single–valued *clusters* $\mathcal{C}$, under the rule that two sites belong to the same cluster if and only if there exists a path between them via a sequence of bonds. We need to determine the clusters from the bonds on each sweep but, although this is quite taxing for large systems, there are standard computational solutions. Finally, for each cluster $\mathcal{C}$, we assign a new binary value to all its components, with the log odds of 1 to 0 being $\sum_{j\in\mathcal{C}} \alpha_j$. This defines the new state $x' \in S$. Note that, in the important special case where $\alpha_i = 0$ for all i, there is complete symmetry between 0's and 1's and each cluster is equally likely to receive the value 0 or 1. Also note that moderately large β_{ij}'s can promote very large clusters and massive changes between x and x', achieving the basic aim.

Ergodicity of the above procedure follows because it is possible, if highly unlikely, that each individual site forms a cluster, allowing any $x' \in S$ to be reached from any previous x. It remains to prove that (60) is satisfied. In fact, we show the stronger Gibbs sampler property between X and R; i.e. that the resulting x' is a draw from the conditional distribution of X

given R, induced by their joint distribution,

$$\begin{aligned}\Pr(X = x,\, R = r) \;&\propto\; \pi(x)\,\nu(r|x) \;\propto\; \prod_i \mathrm{e}^{\alpha_i x_i} \prod_{i<j} (1-p_{ij})^{-1}\, p_{ij}^{r_{ij}}\, (1-p_{ij})^{1-r_{ij}} \qquad (63)\\ &= \prod_i \mathrm{e}^{\alpha_i x_i} \prod_{i<j} (\mathrm{e}^{\beta_{ij} 1[x_i = x_j]} - 1)^{r_{ij}}, \qquad x \in S,\;\; r \in \{0,1\}^{\frac{1}{2}n(n-1)}.\end{aligned}$$

The conditional distribution $\Pr(X = x \,|\, R = r)$ is also proportional to (63), which we now simplify. For any given $R = r$, let $S(r) = \{x \in S : r_{ij} = 1 \Rightarrow x_i = x_j\}$, the set of realizations x that is consistent with r; that is, $\Pr(X = x \,|\, R = r) = 0$ unless $x \in S(r)$. Hence,

$$\Pr(X = x \,|\, R = r) \;\propto\; \prod_i \mathrm{e}^{\alpha_i x_i} \prod_{i<j} (\mathrm{e}^{\beta_{ij}} - 1)^{r_{ij}}, \qquad x \in S(r),$$

and, since the second product does not depend on x, we obtain

$$\Pr(X = x \,|\, R = r) \;\propto\; \prod_i \mathrm{e}^{\alpha_i x_i} \;=\; \prod_{\mathcal{C}} \prod_{j \in \mathcal{C}} \mathrm{e}^{\alpha_j x_j}, \qquad x \in S(r),$$

where, on the right–hand side, the first product is over the clusters $\mathcal{C}$ induced by r and, in the second, the x_j's within the cluster $\mathcal{C}$ all have the same value. The product over $\mathcal{C}$ implies that the value for each cluster is chosen independently and it follows that the conditional distribution of X given R corresponds exactly to the specification of X', given X and R, in the algorithm. Thus, we have verified that the algorithm is a Gibbs sampler between X and R and therefore maintains π. For the original direct proof, see Swendsen and Wang (1987) and, for additional discussion, Edwards and Sokal (1988), Besag and Green (1993) and Fishman (1999), among many others.

Finally, note that, rather than find all of the clusters at each stage, one can instead grow the single cluster $\mathcal{C}$ that corresponds to a randomly chosen site i and then update its components as before. This cluster is therefore chosen from the complete list with probability proportional to its size, which promotes bigger changes. This variation is due to Wolff (1989). For applets that demonstrate both versions, see

`http://rcswww.urz.tu-dresden.de/∼dv857821/SWIM/swimApplet.html`

written by Daniel Vogel in 2003 during a student visit to the University of Washington. Oddly, in statistical applications, it sometimes pays to downweight large clusters by *partial decoupling* (Higdon, 1993)!

Ex. Bayesian inference for the poly–Weibull distribution

For our second example, we refer back to the Bayesian analysis of competing risks models in Section 3.7.1, with $k \geq 2$ components. For the moment, we retain generality, so that the posterior density of the parameters $x = (x_1, \ldots, x_k)$, given the data (y, d), is

$$\pi(x|y,d) \;\propto\; \rho(x) \prod_{i=1}^{n} \bar{F}(y_i|x)\, \{h(y_i|x)\}^{d_i}. \qquad (64)$$

where $\rho(x)$ is the prior for x and the rest of the right–hand side is the likelihood (52). Instead of addressing (64) directly, as in Section 3.7.1, we follow Berger and Sun (1993) in defining additional parameters $z_1, \ldots, z_n$, where

$$z_i = \begin{cases} 0 & \text{if } y_i \text{ is censored} \\ r & \text{if component } r \text{ fails at } y_i \end{cases}$$

Of course, z_i is zero if $d_i = 0$ but is unknown if $d_i = 1$, in which case

$$\Pr(z_i = r \mid x, y_i, d_i = 1) = h_r(y_i|x_r)/h(y_i|x), \qquad r = 1, \ldots, k, \tag{65}$$

from which sampling is trivial. It follows (also from first principles) that the joint posterior density of x and z, given y and d, is

$$\pi(x, z|y, d) \propto \rho(x) \prod_{i=1}^{n} \bar{F}(y_i|x)\, h_{z_i}(y_i|x_{z_i}), \tag{66}$$

with $z_i = 0$ if $d_i = 0$ and $h_0(t|x_0) \equiv 1$. Assuming that $\rho(x) = \rho_1(x_1)\rho_2(x_2)\ldots\rho_k(x_k)$, the full conditional density of x_r is

$$\pi(x_r|x_{-r}, z, y, d) \propto \rho_r(x_r) \prod_{i=1}^{n} \bar{F}_r(y_i|x_r)\, \{h_r(y_i|x_r)\}^{1[z_i=r]}, \tag{67}$$

where $[\ .\ ]$ is the usual indicator function, so that (65) and (67) can be used in constructing an MCMC algorithm for (66). It is evident that, in addition to their possible substantive interpretation, the z_i's play the role of auxiliary variables, as noted by Berger and Sun (1993).

In the particular case of the poly–Weibull distribution, $x_r = (\phi_r, \gamma_r)$ is a vector. Our auxiliary variables implementation then differs from Berger and Sun (1993) in using bivariate Metropolis proposals to update each x_r, rather than considering ϕ_r and γ_r individually and running the corresponding Gibbs sampler. Our approach again simplifies the programming, does not require log–concave full conditionals and allows ordering of the γ_r's as in Section 3.7.1. The results obtained by refitting the bi–Weibull distribution to the rats data agree closely with those obtained previously and are of comparable accuracy for the same run length. Among the 42 uncensored lifetimes, the posterior mean probability that death is attributable to the second component of risk increases monotonically with time of death and is 0.005 for the rat that died at 2 weeks, 0.48 for the death at 70 weeks and 0.85 for the final death at 106 weeks. Note, however, that the same information can be extracted from the PW2MO run in Section 3.7.1 by applying (65) to the x's in the MCMC output.

4.6 Perfect MCMC simulation

Coupling from the past (CFTP) is an MCMC method devised by Propp and Wilson (1996) to produce a *perfect* sample from the target distribution. In effect, CFTP runs the chain from the infinite past and samples it at time zero, so that complete convergence is assured. Although this sounds bizarre, it can be achieved in several important special cases. These include the Ising model (24), even on very large arrays (e.g. 2000 × 2000) and at the most awkward and physically interesting parameter values $\alpha = 0$, $\beta = \beta^*$. Indeed, the random samples of size up to 20000 that we used for the Monte Carlo maximum likelihood example in Section 2.4.2 were generated via CFTP. Recent work has resulted in many extensions of CFTP, including perfect simulation of models with non–denumerable state spaces. The topic is very active: see, among many others, Fill (1998), Foss and Tweedie (1998), Kendall (1998), Murdoch and Green (1998), Propp and Wilson (1998), Häggström and Nelander (1999), Häggström, van Lieshout and Møller (1999), Kendall and Thönnes (1999), Møller (1999a,b), Møller and Schladitz (1999), Thönnes (1999), Burdzy and Kendall (2000), Fill, Machida, Murdoch and Rosenthal (2000) and Wilson (2000). In this section, we describe the basic idea of CFTP and apply it to the posterior distribution (17) for the noisy binary channel. Also, we sketch the motivation behind Murdoch and Green (1998).

Let $\{\pi(x) : x \in S\}$ denote a target distribution, where S is finite. As usual, we consider a Markov t.p.m. P with limiting distribution π but, instead of running forwards m steps from 0, we run the chain forwards from time $-m$, to be determined, and sample at the fixed time 0. Indeed, we now imagine doing this from *every* state $x^{(-m)} \in S$, rather than from a single state, but using the identical stream of random numbers in every case, with the effect that, if any two paths ever enter the same state, then they coalesce permanently. In fact, since S is finite, we can be certain that, if we start the simulation far enough back in the past, coalescence will occur in *all* paths before time 0, so that we obtain the same $x^{(0)}$ for every $x^{(-m)}$. Furthermore, this implies that we would obtain $x^{(0)}$ running the chain from any state in the infinite past, provided we continue to use the identical random number stream during the final m steps, since we know that $x^{(-m)}$ is then irrelevant; and thus $x^{(0)}$ would be a random draw from π. Note that it is crucial to sample at a fixed rather than a random time. Running forwards from time zero, with every possible initialization, and waiting for coalescence of all the paths (Johnson, 1996) produces a *random* stopping time and a corresponding bias in the eventual state. As an extreme example, suppose that $P(x', x'') = 1$ but $P(x, x'') = 0$ for all $x \neq x'$: then $\pi(x'') = \pi(x')$ but coalescence cannot begin in x''.

At first sight, there seems no hope of putting the above ideas usefully into practice. Unless the state space S is tiny, it is not feasible to simulate from every state even for $m = 1$, let alone determine a point sufficiently remote in history that all paths are coincident at time 0. However, the fact that coalescence is permanent suggests that sometimes we may be able to identify extremal states and merely run from these. Thus, for the noisy binary channel, we shall see below that, if $\beta > 0$, it is sufficient merely to ensure coalescence from the "all zeros" and "all ones" states $x = 0$ and $x = 1$ and that this can happen surprisingly fast. The additional property required by Propp and Wilson (1996) is a form of monotonicity in the paths. We discuss this here in the context of our example but the reasoning is identical to that used by Propp and Wilson for the ostensibly much harder Ising model and also extends immediately to the general autologistic model (38), provided the β_{ij}'s are non–negative, which is the case of most common statistical interest.

Ex. Noisy binary channel

We again consider the posterior distribution (17) for the noisy binary channel, with α and $\beta > 0$ known. There is no complication in other forms of independent degradation or in an asymmetric t.p.m., so long as its diagonal elements are dominant. We have seen already that (17) leads to the full conditional distributions (40), so that it is easy to implement a systematic scan Gibbs sampler. In doing so, we presume that the usual inverse distribution function method is employed at every stage: that is, when addressing component x_i, we generate a uniform deviate on the unit interval and, if its value exceeds the probability for $x_i = 0$, implied by (40), we set the new $x_i = 1$, else $x_i = 0$.

Now consider any x', $x'' \in S$ such that $x' \leq x''$, componentwise. Then, this property will be preserved in the next generation, provided we use the same deviates for updating each vector and the inverse distribution function method, with $\beta > 0$. This result can be iterated. Thus, consider initializations by all 0's, by all 1's, and by any other $x \in S$. Since $0 \leq x \leq 1$ componentwise, the corresponding ordering is inherited in each subsequent generation and it follows that all paths must have coalesced by the time the two extreme ones do so. Hence, we need only monitor two paths.

However, we must still determine how far back we need to go. A basic method is as follows. We begin by running simulations from time -1, initialized by $x^{(-1)} = 0$ and $x^{(-1)} = 1$, respectively. If the paths do not coalesce at time 0, we repeat the exercise from time -2, making sure that the previous random numbers are used again between times -1 and 0. If the paths do not coalesce by time 0, we repeat from time -3, ensuring that the previous random numbers are used between times -2 and 0; and so on. The procedure is terminated when coalescence by time 0 occurs, in which case the corresponding $x^{(0)}$ represents a random draw from π. We say "by" rather than "at" time 0 because, in the final run, coalescence may occur before time 0. Incidentally, in practice, it is often more efficient to use increasing increments between the starting times of successive runs. Of course, one must still duplicate the random numbers during the common intervals of any two runs but there is no need to identify the smallest m for which coalescence occurs by time zero, though we did in our example.

For a numerical illustration, we again choose $\alpha = \ln 4$ and $\beta = \ln 3$ in (17), with $y =$ 1110011100..., a vector of length 100000. Thus, the state space has 2^{100000} elements, though recall that the MPM and MAP estimates both coincide with y. Our computer program does not benefit from the repetitive pattern in y. Moving back one step at a time, coalescence by time 0 first occurs when running the algorithm from time -15, which reflects an approximate halving of the discrepancies between each pair of paths, generation by generation, though not even a decrease is guaranteed. Coalescence itself occurs at time -2. There are 77759 matches between y and the CFTP sample $x^{(0)}$, which agrees very closely with the 77710 between y and the sample obtained from the Baum et al. (1970) algorithm. Note that the performance of CFTP depends critically on the parameter values and, not surprisingly, can become hopeless as β increases. In such cases, it may be possible to devise algorithms that converge faster but still preserve monotonicity. Indeed, for the Ising model, Propp and Wilson (1996) replace the Gibbs sampler by Sweeny's (1983) cluster algorithm; Swendsen–Wang cannot be used because it violates the monotonicity condition. Fortunately, in most Bayesian formulations, the information in the likelihood strongly dominates that in the prior and so convergence to π is quite fast.

Murdoch and Green (1998) extend CFTP to continuous state spaces and avoid the need for monotonicity. Below, we merely indicate the underlying idea in the context of a discrete state Markov chain. Thus, let P denote a t.p.m. whose limiting distribution is the row vector π, so that $\pi P = \pi$. Let γ denote the corresponding probability vector whose elements are proportional to the minimal elements in the columns of P: we assume that the minima are not all equal to zero. Let G denote the square matrix, all of whose rows are equal to γ. Then we can write

$$P \;=\; \alpha G + \bar{\alpha} H, \tag{68}$$

where G and H are also t.p.m.'s, $0 < \alpha < 1$ (except in the trivial case where all rows of P are equal) and $\alpha + \bar{\alpha} = 1$. Then one way to update a current state via P is to use G with probability α and H with probability $\bar{\alpha}$. Now note that whenever the choice is G, the current state is irrelevant. Thus, we may run the chain in effect from the infinite past and sample it at time $t = 0$ by generating a final time $t^* < 0$ at which G is chosen and then running forwards to $t = 0$, using G for the first step and H for the remainder. This is simple because $-t^*$ is drawn from a geometric distribution. The validity of the procedure can be confirmed algebraically by calculating the distribution of $X^{(0)}$ as

$$\alpha\gamma \sum_{t=0}^{\infty} (\bar{\alpha} H)^t \;=\; \alpha\gamma (I - \bar{\alpha} H)^{-1} \;=\; \pi,$$

where I is the identity matrix. The existence of the inverse is ensured, because $I - \bar{\alpha} H$ cannot have a zero eigenvalue, and the final equality then holds because $\pi(I - \bar{\alpha} H) = \alpha \pi G$ and $\pi G = \gamma$, since π is a probability vector.

In MCMC, the above assumptions or their analogues in a continuous state space may be violated in several ways. In particular, P or the corresponding kernel may be known only up to scale and may not be rich enough to allow a non–zero α in (68). Murdoch and Green (1998) describe further devices to deal with such problems, at least in low–dimensional examples.

4.7 Reversible jumps MCMC

Another recent advance in MCMC methodology has been the introduction of *reversible jumps* by Green (1995). This provides an important generalization of ideas already employed in spatial statistics for Markov point processes (e.g. Ripley, 1977; Geyer and Møller, 1994), in image analysis (e.g. Grenander and Miller, 1992) and in MCMC multigrid methods (e.g. Sokal, 1989; Besag and Green, 1993) and has become a potent force in Bayesian formulations where it is required to move between parameter spaces of differing dimensions. Thus, in change–point problems, the number of change points can itself be a parameter, allowed to vary stochastically during a single run; and, similarly, in the analysis of mixture distributions (Richardson and Green, 1997), the number of components in the mixture may not be known. Below, we provide a modification (Besag, 1997) of the description in Green (1995), which avoids measure theoretic considerations by equalizing the dimensions of the distributions. Which account one prefers is perhaps a matter of taste.

It is convenient to focus on a specific application and here we choose the original context of spatial point processes. Thus, consider a bounded interval $\mathcal{I}$ on the real line, though there is no complication, so far as the present description is concerned, in considering regions in $\mathcal{R}^d$ or in other spaces. The interval $\mathcal{I}$ is populated by a random number K of points or "stars", as we shall refer to them. Given $K = k$, the stars have random coordinates, which we store in a vector X_k of length k, a slight abuse of our usual notation. Now suppose X_K has a density,

$$\nu(k, x_k) \;=\; \omega(k)\, \nu(x_k|k), \qquad k \in \mathcal{K},\ x_k \in \mathcal{I}^k, \tag{69}$$

where $\mathcal{K} = \{0, 1, \dots k^*\}$, with k^* fixed but arbitrary and ultimately irrelevant. Here $\omega(k)$ is the marginal probability of k stars, $\nu(x_k|k)$ is the conditional density of locations x_k, given k, and $\nu(k, x_k)$ is the density we wish to sample via MCMC. In practice, either $\nu(k, x_k)$ or $\nu(x_k|k)$ may be specified up to scale but only a single scale constant must be present across all k. In the former case, it is unusual for $\omega(k)$ to be available explicitly, because it involves a k–dimensional integral. We assume that $\nu(x_k|k)$ is a genuine k–dimensional density in that multiple stars do not occur at a single location. However, the problem in designing an MCMC algorithm is that we are dealing with distributions of differing dimensions, according to the size of k. Thus, x and x' in equations such as (32) may have different dimensions. Nevertheless, we can proceed as follows.

Let $\{\nu_0(x_k|k) : x_k \in \mathcal{I}^k\}$, for each k, denote some simple k–dimensional density; for example, independent and uniform on $\mathcal{I}$. Now define a "universal" target density $\pi(k, x)$, where $x = (x_0, \dots, x_{k^*})$ stores a pattern for each k, by

$$\pi(k, x) \;=\; \nu(k, x_k) \prod_{l \neq k} \nu_0(x_l|l), \qquad x \in \mathcal{I}^0 \times \dots \times \mathcal{I}^{k^*},\ k \in \mathcal{K}. \tag{70}$$

Note that π has fixed dimension $1 + \frac{1}{2}k^*(k^* + 1)$ and that, if we can sample from it, then the value of k and the corresponding locations x_k are draws from the original target density $\nu(k, x_k)$, marginalizing over the x_l for $l \neq k$.

Thus, our task is now reduced to a conventional one, which we can address via a standard Hastings algorithm for a distribution of fixed dimension. We consider two different types of kernels, proposing a change from the current k and x to a new k' and x': the first proposes a change only in x_k to a new x'_k, whereas the second proposes changes in x_k and $x_{k'}$ to x'_k and $x'_{k'}$ for some $k' \neq k$. In the first case, the proposal will depend only on k and x_k and, in the second, also on k' but not on $x_{k'}$. Further, in the second case, x'_k will be a random draw from $\nu_0(x_k|k)$. It follows that the two types of kernel can be written as $R_k(x_k, x'_k)$ and $\nu_0(x'_k|k)\, R_{kk'}(x_k, x'_{k'})$, respectively. Hence, the quotients in the acceptance probabilities corresponding to (34) reduce to

$$\frac{\nu(k, x'_k)R_k(x'_k, x_k)}{\nu(k, x_k)R_k(x_k, x'_k)}, \tag{71}$$

in the first case, and to

$$\frac{\nu(k', x'_{k'})R_{k'k}(x'_{k'}, x_k)}{\nu(k, x_k)R_{kk'}(x_k, x'_{k'})}, \tag{72}$$

in the second, all other terms cancelling out. Consequently, we need never store $x_{k'}$ for any k' other than the current k, nor ever generate x'_k from $\nu_0(x_k|k)$, nor even specify the ν_0's since they never need to be used! Also k^* is irrelevant.

Note that extra care is required to ensure the validity of the proposal mechanism when changing dimension. In point process applications, it is usual to allow only three types of proposal, in which (i) $k' = k - 1$, (ii) $k' = k + 1$ or (iii) $k' = k$. Typically, the proposal is obtained in (i) by deleting a randomly chosen star from the current x_k; in (ii) by adding a star to x_k, with location e.g. uniform in $\mathcal{I}$; in (iii) by moving a randomly chosen star to a uniformly chosen location. If the proposal is rejected, we obtain (iv) in which there is no change. However, note that it would not be legitimate in (i) to select two stars at random and replace them by a single star located at their centroid, because (ii) would not allow the reverse move. This is clear but, more subtly, nor would it be legitimate to additionally revise (ii) to delete a star at random and add two more whose locations are uniform on $\mathcal{I}$. That is, the cancellations that we have just seen must occur genuinely, also with respect to the proposal distributions. For further discussion, see the elegant measure theoretic description in Green (1995) and also the comments in the example below. For further details of point process simulation, see, for example, Geyer and Møller (1994) and Häggström et al. (1999).

Ex. Bayesian inference for the poly–Weibull distribution

In Section 3.7.1, we discussed Bayesian inference for data from a censored poly–Weibull distribution having a fixed number k of components. We saw that a very simple Metropolis algorithm provides a satisfactory and more flexible alternative to the somewhat daunting Gibbs sampler described by Berger and Sun (1993) and briefly discussed the fit of the model with $k = 1, 2$ and 3 to the same data as in Davison and Louzada–Neto (2000). Now we extend the formulation by allowing $k = 1, 2$ and 3 within a single run. This requires a prior distribution $\omega(k)$ for k, though one can always make a notional choice and subsequently reweight the results appropriately. Indeed, a token prior, chosen to encourage good mixing, is often preferable computationally. Here, we take $\omega(k) = 0.9, 0.05, 0.05$ for $k = 1, 2, 3$. Incidentally, larger values of k do not seem warranted for the data at hand.

We begin with some minor alterations in notation, in addition to the reinterpretation of x_k and x. Thus, $\nu(k, x_k)$ and $\pi(k, x)$ in (69) and (70) become posterior densities,

$$\begin{aligned} \nu(k, x_k|y) &\propto L(y|k, x_k)\, \nu(x_k|k)\, \omega(k), \\ \pi(k, x|y) &= \nu(k, x_k|y) \prod_{l \neq k} \nu_0(x_l|l), \end{aligned}$$

conditioned by the observed data vector y, which now also includes the information d about censoring. The likelihood $L(y|k, x_k)$ is the same as in Section 3.7.1 but the notation accommodates a varying number k of components in the poly–Weibull formulation. The restriction to $k = 1, 2, 3$ is convenient in the description but loses almost nothing in generality and is not required in our computer program. Thus, we omit x_0 and define

$$x_1 = x_{11}, \qquad x_2 = (x_{21}, x_{22}), \qquad x_3 = (x_{31}, x_{32}, x_{33}),$$

where $x_{kl} = (\phi_{kl}, \gamma_{kl})$, $\phi_{kl} = \ln \theta_{kl}$ and $\gamma_{kl} = \ln \beta_{kl}$. For definiteness, we only consider ordered parametrizations here, with $\gamma_{21} \leq \gamma_{22}$ and $\gamma_{31} \leq \gamma_{32} \leq \gamma_{33}$. This means that we can reference *adjacent* pairs of components x_{kl} and x_{kl+1}, though here this is relevant only when $k = 3$.

For any particular k, we again adopt the Davison and Louzada–Neto (2000) prior for the ϕ_r's and γ_r's, subject to the required ordering. Note that the ordering implies that the right–hand side of (53) should be multiplied by $k!$ and here this matters because we allow k to vary. We again set $a_r = 100$ for any k and r. The posterior density $\nu(k, x_k|y)$ is then proportional to $k!\,\omega(k)$ times the right–hand side of equation (54). It is crucial here that the normalizing constant is independent of k, for which the previously irrelevant term $\prod_r a_r = 100^k$ must also be included. Note that our choice of prior is not entirely self–consistent when k is a parameter of the formulation and we view our analysis as primarily illustrative. Of course, any other choice can be made, so long as the normalizing constant is known.

We allow four types of transition, corresponding to (i), (ii), (iii) and (iv) for point processes (see Section 4.5) and which can be thought of as *merges*, *splits*, *walks* and *rests*, respectively. Thus, in (i), two adjacent pairs of components are merged in some manner into a single pair; in (ii), a single pair of components splits in some way into two adjacent pairs; in (iii), the number of components remains the same but their values are all changed; and, in (iv), there are no changes at all, which occurs if a merge, split or walk is rejected for any reason. The values of other components in (i) and (ii) are carried forward into the new $x'_{k'}$, as we exemplify below. At every stage, the required ordering must be preserved and any proposal that violates it is immediately rejected. This is not the most efficient scheme but it suffices for now. Of course, in defining (iii), we could choose to change some, rather than all, components.

Each cycle of the algorithm proceeds as follows. For the current k and corresponding x_k, we first choose the type of proposal to be made: if $k = 1$, a split with probability $\frac{1}{3}$, else a walk; if $k = 2$, a merge or a split of a randomly selected pair or a walk, each with probability $\frac{1}{3}$; if $k = 3$, a merge of a randomly selected adjacent pair with probability $\frac{1}{3}$, else a walk. For example, if $k = 3$, we propose merging x_{31} and x_{32} to form x'_{21} and carrying over $x'_{22} = x_{33}$, with probability $\frac{1}{6}$. If, in the proposed merge, $\gamma'_{21} > \gamma'_{22}$, we immediately reject it and take a rest; that is, retain $k' = 3$ and $x'_3 = x_3$. Similarly, if $k = 2$, we propose splitting x_{21} to form x'_{31} and x'_{32} and carrying forward $x'_{33} = x_{22}$, again with probability $\frac{1}{6}$. If the proposed split violates $\gamma'_{31} < \gamma'_{32} < \gamma'_{33}$, we again reject it and take a rest.

It remains to define the exact meaning of merges, splits and walks and to determine the acceptance probabilities for the corresponding proposals. For walks, we propose new ϕ_r's and γ_r's exactly as in Section 3.7.1 and these are accepted or rejected as in the WeibM, PW2MO and PW3MO algorithms, for $k = 1$, 2 and 3, respectively. It is when we consider merges and splits that we perhaps best see a difference between our general description of reversible jumps in Section 4.5 and those in Green (1995), Richardson and Green (1997) and elsewhere. Without any loss of generality, we return to the examples in the previous paragraph. Then, in merging x_{31} and x_{32}, we define the proposal x'_{21} by

$$\phi'_{21} = \tfrac{1}{2}(\phi_{31} + \phi_{32}) + z_\phi, \qquad \gamma'_{21} = \tfrac{1}{2}(\gamma_{31} + \gamma_{32}) + z_\gamma, \tag{73}$$

where z_ϕ and z_γ are independent Gaussian variates with zero means and prescribed standard deviations, which we choose to be 0.5 in our numerical example. Correspondingly, in splitting x_{21}, we define the proposals x'_{31} and x'_{32} by

$$\phi'_{31} = \phi_{21} + z_{\phi_1}, \qquad \gamma'_{31} = \gamma_{21} + z_{\gamma_1}, \qquad \phi'_{32} = \phi_{21} + z_{\phi_2}, \qquad \gamma'_{32} = \gamma_{21} + z_{\gamma_2}, \tag{74}$$

where the z_ϕ's and z_γ's are also independent Gaussian variates with zero means and prescribed standard deviations, which we again take as 0.5 in our example. These proposals are complementary and it follows that the Hastings ratio for a proposed merge from x_3 to x'_2, with $x'_{22} = x_{33}$, is

$$\frac{\nu(2, x'_2|y) f_{23}(x'_{21}; x_{31}, x_{32})}{\nu(3, x_3|y) f_{32}(x_{31}, x_{32}; x'_{21})},$$

where $f_{32}(.;.)$ represents the two–dimensional density for the merge and $f_{23}(.;.)$ the four–dimensional density for the split, described above, except that the primes are transferred from the left–hand sides to the right–hand sides of (74). Correspondingly, the ratio for a proposed split from x_2 to x'_3, with $x'_{33} = x_{22}$, is

$$\frac{\nu(3, x'_3|y) f_{32}(x'_{31}, x'_{32}; x_{21})}{\nu(2, x_2|y) f_{23}(x_{21}; x'_{31}, x'_{32})},$$

where the densities are for the above proposals, except that the primes are transferred in equations (73). Note the cancellation of an additional factor $\frac{1}{6}$ in each numerator and denominator, because of the way in which the move types are chosen.

The above description extends immediately to produce a complete algorithm. Thus, for the data analyzed by Davison and Louzada–Neto (2000), we find that the Bayes factor in favour of $k = 2$ over $k = 1$ is about 55 but for $k = 3$ over $k = 2$ is only about 1.2. Recall that, although we need a token prior for k to run the MCMC, subsequent rescaling should produce a Bayes factor that does not depend on the particular choice. Indeed, we refrain here from making any inferences that depend explicitly on the prior. Of course, we can also extract information conditional on k and verify that the results agree with those from separate runs in Section 3.7.1.

Returning to methodology, our description applies also to a general k^*, rather then merely $k^* = 3$ and, indeed, with appropriate modifications, to many other applications of reversible jumps. One can easily relax the very harsh rejection rule: for example, in (74), one might first order the two (ϕ, γ)–pairs but one must then be careful to take account of multiple paths in the Hastings ratio. One can also abandon ordering altogether.

Finally, we extract a more common form of merge and split partnership for mixture models from our formulation. Thus, first consider (73). In practice, it would often be appealing to assign very small values to the associated standard deviations, so that ϕ'_{21} and γ'_{21} fall virtually half way between the corresponding pairs of current parameter values. Then, in order to maintain moderate acceptance probabilities, we could replace the independent z_ϕ's in (74) by ones that are highly negatively correlated and indeed almost equal but opposite; and similar considerations apply to the z_γ's. If we take this choice to its logical conclusion, we obtain, instead of (73),

$$\phi'_{21} = \tfrac{1}{2}(\phi_{31} + \phi_{32}), \qquad \gamma'_{21} = \tfrac{1}{2}(\gamma_{31} + \gamma_{32})$$

and, instead of (74),

$$\phi'_{31} = \phi_{21} + z^*_\phi, \qquad \gamma'_{31} = \gamma_{21} + z^*_\gamma, \qquad \phi'_{32} = \phi_{21} - z^*_\phi, \qquad \gamma'_{32} = \gamma_{21} - z^*_\gamma,$$

where z^*_ϕ and z^*_γ have prescribed standard deviations. These are precisely the sorts of proposals that are typical in multigrid MCMC algorithms for continuous variables (see, for example, Sokal, 1989) and that are adopted in Green (1995) and Richardson and Green (1997) for simulating posterior distributions in mixture models. Note that there is a slight complication because the above and any similar transformations acquire a simple Jacobian that appears in the Hastings acceptance ratio; see equations (7) and (8) in Green (1995). The equivalent result can be derived from (73) and (74) via an appropriate limiting argument. Our general approach separates the issues concerning changes of dimensions that necessarily arise in merges and splits from those that occur when one chooses deliberately to adopt singular proposal distributions in the implementation itself.

4.8 Simulated annealing

We briefly recall the basic ideas from Section 2.5. Thus, let $\{h(x) : x \in S\}$, with S finite, denote a bounded non–negative function, specified at least up to scale. Then the goal is to locate the optimal value $x^+ = \arg\max_x h(x)$ of x, assuming this is beyond the capabilities of more conventional methods. Simulated annealing was introduced for such tasks by Kirkpatrick, Gelatt and Vecchi (1983). The earliest statistical applications were to Bayesian image analysis (Geman and Geman, 1984) and to optimal experimental design (Haines, 1987). More generally, simulated annealing is relevant to any finite maximization arising in decision theory. For corresponding algorithms in continuous spaces, using Langevin diffusion, see Geman and Hwang (1986). However, our illustrative example below is not from statistics but from operations research, where discrete optimization problems abound. Although it is often possible to design deterministic algorithms that are more efficient in solving particular tasks, simulated annealing is remarkably successful across a wide range of applications, especially in high dimensions.

Instead of a single target distribution, simulated annealing addresses a whole family of distributions

$$\pi_k(x) \;\propto\; \{h(x)\}^{m_k}, \qquad x \in S,$$

for $k = 1, 2, \ldots$, where the m_k's form a sequence of increasing positive constants, with $m_1 = 1$, say. Clearly, each π_k has its mode at x^+ and, as k increases, π_k becomes progressively more concentrated on x^+; or uniformly on the x^+'s if there are multiple global modes. The aim in simulated annealing is to design an MCMC algorithm that samples the dynamically changing distributions π_k and therefore eventually reaches x^+. This requires that the m_k's increase rather slowly, especially in the latter stages of the algorithm, because the observation attributed to π_k must also serve approximately as a draw from π_{k+1}. The advantage of a stochastic algorithm is that it permits escapes from local modes as a matter of course, especially in the earlier stages. It may seem counter–intuitive that such a scheme can succeed in practice with limited computational resources but the hope is that eventually a state is reached that is acceptably close to x^+.

Simulated annealing takes its name by analogy with physical annealing in which hot steel is cooled slowly as a means of toughening. Thus, we can interpret $\tau_k = 1/m_k$ as "temperature" and, with this in mind, the corresponding sequence of τ_k's is usually referred to as a *temperature schedule*. This schedule is crucial to the performance of the algorithm. Theory suggests that the m_k's should increase at a rate closer to logarithmic than linear for total success but, in practice, it is impracticable to adhere to such a severe schedule.

Ex. Traveling salesman problem

Suppose that a salesperson, located at 0, must visit each of n fixed locations $i = 1, 2, \ldots, n$ once before returning to base. The travel time v_{ij} from each i to each j is known and the task is to minimize the total round–trip time. In our example below, $v_{ji} \equiv v_{ij}$ but this is not always the case. The seminal paper by Dantzig, Fulkerson and Johnson (1954) provides a deterministic solution to the traveling salesman problem (TSP) via integer linear programming but this can run into problems if n is too large. TSP is an NP–complete problem.

For a simulated annealing algorithm, let x denote a valid route, which we represent by a string $0, r_1, \ldots, r_n, 0$, where $r_1, \ldots, r_n$ is any permutation of the integers $1, \ldots, n$. Let S denote the set of all $n!$ allowable routes and define a family of distributions,

$$\pi_k(x) \ \propto \ \exp\{-m_k v(x)\}, \qquad x \in S,$$

where $v(x)$ is the round–trip time for route $x \in S$. The most naive Metropolis algorithm for π_k is based on proposals x^* that differ from the current state x by the transposition of two randomly selected indices r_i and r_j, say. The corresponding acceptance probability is then 1 if $v(x^*) \leq v(x)$ and $\exp[-m_k\{v(x^*) - v(x)\}]$ if $v(x^*) > v(x)$. It remains to choose the sequence of m_k's.

For a toy numerical example, suppose that $n = 99$ and that the 100 locations are the vertices of a 10×10 square array with unit spacing. We measure the travel time v_{ij} between locations i and j by their Euclidean distance apart. The labeling of the vertices is irrelevant but for definiteness we adopt raster scan with vertex 0 at the top left of the array. Obviously, the minimal travel time is 100 and can be achieved in many different ways, which simplifies the task. We tried several different temperature schedules and eventually used a run length of 4 billion, with m_k increasing linearly from 1 to 100, to attain a perfect route. Travel times of less than 102 occurred for much shorter runs,

Not surprisingly, the performance of simulated annealing in locating x^+ is highly context dependent. The technique is quite popular in Bayesian image analysis, where x^+ is the MAP estimate of the true image but the results shown are often rather far removed from the actual x^+ and are sometimes more impressive! For examples and discussion of this apparent paradox, see Marroquin, Mitter and Poggio (1987) and Greig, Porteous and Seheult (1989).

References

Amit, Y., Grenander, U. and Piccioni, M. (1991). Structural image restoration through deformable templates. *Journal of the American Statistical Association,* **86,** 376–387.

Baddeley, A. J. (2000). Time–invariance estimating equations. *Bernoulli,* **6,** 1–26.

Barnard, G. A. (1963). Discussion of paper by M. S. Bartlett. *Journal of the Royal Statistical Society B,* **25,** 294.

Bartolucci, F. and Besag, J. E. (2002). A recursive algorithm for Markov random fields. *Biometrika,* **89,** 724–730.

Baum, L. E., Petrie, T., Soules, G. and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics,* **41,** 164–171.

Berger, J. O. and Sun, D. (1993). Bayesian analysis for the poly–Weibull distribution. *Journal of the American Statistical Association,* **88,** 1412–1417.

Besag, J. E. (1974). Spatial interaction and the statistical analysis of lattice systems (with Discussion). *Journal of the Royal Statistical Society B,* **36,** 192–236.

Besag, J. E. (1975). Statistical analysis of non–lattice data. *The Statistician,* **24,** 179–195.

Besag, J. E. (1978). Some methods of statistical analysis for spatial data (with Discussion). *Bulletin of the International Statistical Institute,* **47,** 77–92.

Besag, J. E. (1989). Towards Bayesian image analysis. *Journal of Applied Statistics,* **16,** 395–407.

Besag, J. E. (1992). Simple Monte Carlo p-values. In *Proceedings of Interface 90* (eds. C. Page and R. LePage), 158–162. Springer–Verlag: New York.

Besag, J. E. (1994). Discussion of paper by U. Grenander and M. I. Miller. *Journal of the Royal Statistical Society B,* **56,** 591–592.

Besag, J. E. (1997). Discussion of paper by S. Richardson and P. J. Green. *Journal of the Royal Statistical Society B,* **59,** 774.

Besag, J. E. and Clifford, P. (1989). Generalized Monte Carlo significance tests. *Biometrika,* **76,** 633–642.

Besag, J. E. and Clifford, P. (1991). Sequential Monte Carlo *p*–values. *Biometrika,* **78,** 301–304.

Besag, J. E. and Diggle, P. J. (1977). Simple Monte Carlo tests for spatial pattern. *Applied Statistics,* **26,** 327–333.

Besag, J. E. and Green, P. J. (1993). Spatial statistics and Bayesian computation (with Discussion). *Journal of the Royal Statistical Society B,* **55,** 25–37.

Besag, J. E., Green, P. J., Higdon, D. M. and Mengersen, K. L. (1995). Bayesian computation and stochastic systems (with Discussion). *Statistical Science,* **10,** 3–66.

Besag, J. E. and Higdon, D. M. (1999). Bayesian analysis of agricultural field experiments (with Discussion). *Journal of the Royal Statistical Society B,* **61,** 691–746.

Besag, J. E. and Mondal D. (2004). Exact *p*–values for Markov chains. To appear.

Besag, J. E. and Tantrum, J. (2003). Likelihood analysis of binary data in space and time. In *Highly Structured Stochastic Systems* (eds. P. J. Green, N. L. Hjort and S. Richardson), 289–295. Oxford University Press.

Besag, J. E., York, J. C. and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics (with Discussion). *Annals of the Institute of Statistical Mathematics,* **43,** 1–59.

Bishop, Y. M. M., Fienberg, S. E. and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice.* M.I.T. Press: Cambridge.

Bunea, F. and Besag, J. E. (2000). MCMC in $I \times J \times K$ contingency tables. In *Monte Carlo Methods* (ed. N. Madras), *Fields Institute Communications,* **26,** 25–36.

Bøvliken, E. and Skovlund, E. (1996). Confidence intervals from Monte Carlo tests. *Journal of the American Statistical Association,* **91,** 1071–1078.

Burdzy, K. and Kendall, W. S. (2000). Efficient Markovian couplings: examples and counterexamples. *Annals of Applied Probability,* **10,** 362–409.

Byers, S. D. and Besag, J. E. (2000). A geographical analysis of prostatic cancer in the USA, involving ethnicity. *Statistics in Medicine,* to appear.

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

Cox, D. R. and Wermuth, N. (1994). A note on the quadratic exponential binary distribution. *Biometrika,* **81,** 403–408.

Creutz, M. (1979). Confinement and the critical dimensionality of space-time. *Physics Review Letters,* **43,** 553–556.

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

Dantzig, G. B., Fulkerson, D. R. and Johnson, S. (1954). Solution of a large–scale traveling salesman problem. *Journal of the Operations Research Society of America,* **2,** 393–410.

Davison, A. C. and Louzada–Neto, F. (2000). Inference for the poly–Weibull distribution. *The Statistician,* **49,** 189–196.

Diaconis, P. and Saloff-Coste, L. (1993). Comparison theorems for reversible Markov chains. *Annals of Applied Probability,* **3,** 696–730.

Diaconis, P. and Stroock, D. (1991). Geometric bounds for eigenvalues of Markov chains. *Annals of Applied Probability,* **1,** 36–61.

Diaconis, P. and Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distributions. *Annals of Statistics,* **26,** 363–398.

Diggle, P. J. (1983). *Statistical Analysis of Spatial Point Patterns.* Academic Press: London.

Doucet, A., de Freitas, N. and Gordon, N. (eds.) (2001). *Sequential Monte Carlo Methods in Practice.* Springer–Verlag: New York.

Dwass, M. (1957). Modified randomization tests for nonparametric hypotheses. *Annals of Mathematical Statistics,* **28,** 181–187.

Eddie, S. R., Mitchison, G. and Durbin, R. (1995). Maximum discrimination hidden Markov models of sequence concensus. *Journal of Computational Biology,* **2,** 9–24.

Edwards, R. G. and Sokal, A. D. (1988). Generalization of the Fortuin–Kasteleyn–Swendsen–Wang representation and Monte Carlo algorithm. *Physics Review D* **38,** 2009–2012.

Fill, J. A. (1998). An interruptible algorithm for perfect sampling via Markov chains. *Annals of Applied Probability,* **8,** 131–162.

Fill, J. A., Machida, M., Murdoch, D. J. and Rosenthal, J. S. (2000). Extensions of Fill's perfect rejection sampling algorithm to general chains. In *Monte Carlo Methods* (ed. N. Madras), *Fields Institute Communications,* **26,** 37–52.

Fishman, G. S. (1996). *Monte Carlo: Concepts, Algorithms, and Applications.* Springer–Verlag: New York.

Fishman, G. S. (1999). An analysis of Swendsen–Wang and related sampling methods. *Journal of the Royal Statistical Society B,* **61,** 623–641.

Forster, J. J., McDonald, J. W. and Smith, P. W. F. (1996). Monte Carlo exact conditional tests for log-linear and logistic models. *Journal of the Royal Statistical Society B,* **58,** 445–453.

Foss, S. G. and Tweedie, R. L. (1998). Perfect simulation and backward coupling. *Stochastic Models,* **14,** 187–203.

Fredkin, D. R. and Rice, J. A. (1992). Maximum likelihood estimation and identification directly from single–channel recordings. *Proceedings of the Royal Society of London B,* **249,** 125–132.

Gamerman, D. (1997). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference.* Chapman and Hall: London.

Gelfand, A. E., Hills, S. E., Racine–Poon, A. and Smith, A. F. M. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association,* **85,** 972–985.

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

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1995). *Bayesian Data Analysis.* Chapman and Hall/CRC: Boca Raton.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *Institute of Electrical and Electronics Engineers, Transactions on Pattern Analysis and Machine Intelligance,* **6,** 721–741.

Geman, S. and Hwang, C.–R. (1986). Diffusions for global optimization. *SIAM Journal on Control and Optimization,* **24,** 1031–1043.

Geman, S. and McClure, D. E. (1986). Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute,* **52,** 5–21.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (ed. E. M. Keramidas), 156–163. Interface Foundation of North America, Fairfax Station, VA.

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

Geyer, C. J. and Møller, J. (1994). Simulation procedures and likelihood inference for spatial point processes. *Scandanavian Journal of Statistics,* **21,** 84–88.

Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data (with Discussion). *Journal of the Royal Statistical Society B,* **54,** 657–699.

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association,* **90,** 909–920.

Gilks, W. R. (1992). Derivative-free adaptive rejection sampling for Gibbs sampling. In *Bayesian Statistics 4* (eds. J. O. Berger, J. M. Bernardo, A. P. Dawid and A. F. M. Smith), 641–649. Oxford Univ. Press.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. (eds.) (1996). *Markov Chain Monte Carlo in Practice.* Chapman and Hall: London.

Gotelli, N. J. and Entsminger, G. L. (2001). Swap and fill algorithms in null model analysis: rethinking the knight's tour. *Oecologia,* **129,** 281–291.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika,* **82,** 711–732.

Greig, D. M., Porteous, B. M. and Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society B,* **51,** 271–279.

Grenander, U. (1983). Tutorial in pattern theory. Report: Division of Applied Mathematics, Brown University.

Grenander, U. and Miller, M. I. (1994). Representations of knowledge in complex systems (with Discussion). *Journal of the Royal Statistical Society B,* **56,** 549–603.

Guo, S. W. and Thompson, E. A. (1994). Monte Carlo estimation of mixed models for large complex pedigrees. *Biometrics,* **50,** 417–432.

Häggström, O. and Nelander, K. (1999). On exact simulation of Markov random fields using coupling from the past. *Scandanavian Journal of Statistics,* **26,** 395–411.

Häggström, O., van Lieshout M. N. M. and Møller, J. (1999). Characterization results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. *Bernoulli,* **5,** 641–658.

Haines, L. M. (1987). The application of the annealing algorithm to the construction of exact optimal designs for linear–regression models. *Technometrics,* **29,** 439–447.

Hall, P. and Titterington, D. M. (1989). The effect of simulation order on level acuracy and power of Monte Carlo tests. *Journal of the Royal Statistical Society B,* **51,** 459–467.

Hammersley, J. M. and Clifford, P. (1971). Markov fields on finite graphs and lattices. Unpublished manuscript.

Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo methods.* Wiley: London

Hammersley, J. M. and Morton, K. W. (1954). Poor man's Monte Carlo. *Journal of the Royal Statistical Society B,* **16,** 23–38.

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

Haussler, D., Krogh, A., Mian, S. and Sjolander, K. (1993). Protein modeling using hidden Markov models: analysis of globins. In *Proceedings of the Hawaii International Conference on System Sciences.* IEEE Computer Science Press: Los Alamitos, CA.

Higdon, D. M. (1993). Discussion of meeting on MCMC methods. *Journal of the Royal Statistical Society B,* **55,** 78.

Higdon, D. M. (1994). Ph.D. thesis. University of Washington.

Higdon, D. M. (1998). Auxiliary variables methods for Markov chain Monte Carlo with applications. *Journal of the American Statistical Association,* **93,** 585–595.

Hinton, G. E. and Sejnowski, T. (1986). Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing* (eds. D. E. Rumelhart and J. L. McClelland). M.I.T Press.

Hughes, J. P., Guttorp, P. and Charles, S. P. (1999). A nonhomogeneous hidden Markov model for precipitation. *Applied Statistics,* **48,** 15–30.

Jöckel, K.–H. (1986). Finite–sample properties and asymptotic efficiency of Monte Carlo tests. *Annals of Statistics,* **14,** 336–347.

Johnson, V. E. (1996). Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *Journal of the American Statistical Assocociation,* **91,** 154–166.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. and Saul, L. K. (1998). An introduction to variational methods for graphical models. In *Learning in Graphical Models* (ed. M. I. Jordan). Kluwer Academic Publishers.

Juang, B. H. and Rabiner, L. R. (1991). Hidden Markov models for speech recognition. *Technometrics,* **33,** 251–272.

Kendall, W. S. (1998). Perfect simulation for the area–interaction point process. In *Probability Towards 2000* (eds. C. C. Heyde and L. Accardi). Springer–Verlag.

Kendall, W. S. and Thönnes, E. (1999). Perfect simulation in stochastic geometry. *Pattern Recognition,* **32,** 1569–1586.

Kipnis, C. and Varadhan, S. R. S. (1986). Central limit theorem for additive functionals of reversible Markov processes and applications to simple exclusions. *Communications in Mathematical Physics,* **104,** 1–19.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science,* **220,** 671–680.

Knorr–Held, L. and Besag, J. E. (1998). Modelling risk from a disease in time and space. *Statistics in Medicine,* **17,** 2045–2060.

Lagakos, S. W. and Louis, T. A. (1988). Use of tumour lethality to interpret tumorigenicity experiments lacking cause–of–death data. *Applied Statistics,* **37,** 169–179.

Lazzeroni, L. C. and Lange, K. (1997). Markov chains for Monte Carlo tests of genetic equilibrium in multidimensional contingency tables. *Annals of Statistics,* **25,** 138–168.

Le Strat, Y. and Carrat, F. (1999). Monitoring epidemiologic surveillance data using hidden Markov models. *Statistics in Medicine,* **18,** 3463–3478.

Liggett, T. M. (1999). *Interacting Particle Systems.* Springer–Verlag.

Liu, J. S. (1996). Peskun's theorem and a modified discrete–state Gibbs sampler. *Biometrika,* **83,** 681–682.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing.* Springer–Verlag: New York.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association,* **93,** 1032–1044.

Liu, J. S., Neuwald, A. F. and Lawrence, C. E. (1995). Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *Journal of the American Statistical Association,* **90,** 1156–1170.

MacCormick, J. and MacCormick, F. (2002). *Stochastic Algorithms for Visual Tracking: Probabilistic Modelling and Stochastic Algorithms for Visual Localisation and Tracking.* Springer–Verlag: New York.

MacDonald, I. L. and Zucchini, W. (1997). *Hidden Markov and Other Models for Discrete–valued Time Series.* Chapman and Hall: London.

Maitra, R. and Besag, J. E. (1998). Bayesian reconstruction in synthetic magnetic resonance imaging. In *Bayesian Inference in Inverse Problems* (ed. A. Mohammad–Djafari). *Proceedings of SPIE 1998,* **3459,** 39–47.

Manly, B. F. J. (1995). A note on the analysis of species co–occurrences. *Ecology,* **76,** 1109–1115.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters,* **19,** 451–458.

Marroquin, J., Mitter, S. and Poggio, T. (1987). Probabilistic solution if ill–posed problems in computer vision. *Journal of the American Statistical Association,* **82,** 76–89.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics,* **21,** 1087–1092.

Meyn, S. P. and Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability.* Springer–Verlag: London.

Mira, A. and Roberts, G. O. (2003). Discussion of paper by R. M. Neal. *Annals of Statistics,* **31,** 748–753.

Moffett, J. L., Besag, J. E., Byers, S. D. and Li, W.–H. (1997). Probabilistic classification of forest structures by hierarchical modelling of the remote sensing process. *Proceedings of SPIE International Symposium on Optical Science, Engineering and Instrumentation, San Diego.* To appear.

Møller, J. (1999a). Perfect simulation of conditionally specified models. *Journal of the Royal Statistical Society B,* **61,** 251–264.

Møller, J. (1999b). *Aspects of Spatial Statistics, Stochastic Geometry and Markov Chain Monte Carlo Methods.* Unpublished D.Sc. thesis. Faculty of Engineering and Science, Aalborg University.

Møller, J. and Schladitz, K. (1999). Extensions of Fill's algorithm for perfect simulation. *Journal of the Royal Statistical Society B,* **61,** 955–969.

Møller, J., Syversveen, A. R. and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandanavian Journal of Statistics,* **25,** 451–482.

Murdoch, D. J. and Green, P. J. (1998). Exact sampling from a continuous state space. *Scandanavian Journal of Statistics,* **25,** 483–502.

Neal, R. M. (2003). Slice sampling. *Annals of Statistics,* **31,** 705–767.

Newman, M. E. J. and Barkema, G. T. (1999). *Monte Carlo Methods in Statistical Physics.* Clarendon Press: Oxford.

Nummelin, E. (1984). *General Irreducible Markov Chains and Non–Negative Operators.* Cambridge University Press.

Patefield, W. M. (1981). Algorithm AS 159. An efficient method of generating random $r \times c$ tables with given row and column tables. *Applied Statistics,* **30,** 91–97.

Penttinen, A. (1984). Modelling interaction in spatial point patterns: parameter estimation by the maximum likelihood method. *Jyväskylä Studies in Computer Science, Economics and Statistics,* **7.**

Peskun, P. H. (1973). Optimum Monte Carlo sampling using Markov chains. *Biometrika,* **60,** 607–612.

Propp, J. G. and Wilson, B. M. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms,* **9,** 223–252.

Propp, J. G. and Wilson, B. M. (1998). How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree to a directed graph. *Journal of Algorithms,* **27,** 170–217.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the Institute of Electrical and Electronics Engineers,* **77,** 257–284.

Rasch, G. (1960). *Probabilistic Models for some Intelligence and Attainment Tests.* Danish Educational Research Institute: Copenhagen.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with Discussion). *Journal of the Royal Statistical Society B,* **59,** 731–792.

Ripley, B. D. (1977). Modelling spatial patterns (with Discussion). *Journal of the Royal Statistical Society B,* **39,** 172–212.

Ripley, B. D. (1979). Algorithm AS 137: simulating spatial patterns: dependent samples from a multivariate density. *Applied Statistics,* **28,** 109–112.

Robert, C. P., Rydén, T. and Titterington, D. M. (2000). Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method. *Journal of the Royal Statistical Society B,* **62,** 57–75.

Robert, C. P., Rydén, T. and Titterington, D. M. (1998). Convergence controls for MCMC algorithms, with applications to hidden Markov chains. *Journal of Statistical Computation and Simulation,* **64,** 327–356.

Roberts, G. O. and Rosenthal, J. S. (1999). Convergence of slice sampler Markov chains. *Journal of the Royal Statistical Society B,* **61,** 643–660.

Roberts, G. O. and Tweedie, R. L. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika,* **83,** 95–110.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of Langevin diffusions and their discrete approximations. *Bernoulli,* **2,** 3341–363.

Rosenbluth, M. N. and Rosenbluth, A. W. (1955). Monte Carlo calculations of the average extension of molecular chains. *Journal of Chemical Physics,* **23,** 356–359.

Sanderson, J. G. (2000). Testing ecological patterns. *American Scientist,* **88,** 332–339.

Sanderson, J. G., Moulton, M. P. and Selfridge, R. G. (1998). Null matrices and the analysis of species co–occurrences. *Oecologia,* **116,** 275–283.

Smith, P. W. F., Forster, J. J. and McDonald, J. W. (1996). Monte Carlo exact tests for square contingency tables. *Journal of the Royal Statistical Society A,* **159,** 309–321.

Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with Discussion). *Journal of the Royal Statistical Society B,* **55,** 39–52.

Sokal, A. D. (1989). Monte Carlo methods in statistical mechanics: foundations and new algorithms. *Cours de Troisième Cycle de la Physique en Suisse Romande,* Lausanne.

Stone, L. and Roberts, A. (1990). The checkerboard score and species distributions. *Oecologia,* **85,** 74–79.

Suomela, P. (1976). Ph.D. thesis. University of Jyväskylä, Finland.

Sweeny, M. (1983). Monte Carlo study of weighted percolation clusters relevant to the Potts model. *Physical Review B,* **27,** 4445–4455.

Swendsen, R. H. and Wang, J.-S. (1987). Non-universal critical dynamics in Monte Carlo simulations. *Physics Review Letters,* **58,** 86–88.

Thönnes, E. (1999). Perfect simulation of some point processes for the impatient user. *Advances in Applied Probability, 31,* 69–87.

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

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association,* **81,** 82–86.

Tjelmeland, H. and Besag, J. (1998). Markov random fields with higher–order interactions. *Scandanavian Journal of Statistics,* **25,** 415–433.

Weir, I. S. (1997). Fully Bayesian reconstruction from single–photon emission computed tomography data. *Journal of the American Statistical Association,* **92,** 49–60.

Wilson, D. B. (2000). Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP). In *Monte Carlo Methods* (ed. N. Madras), *Fields Institute Communications,* **26,** 143–179.

Winkler, G. (2003). *Image Analysis, Random Fields and Markov chain Monte Carlo Methods: a Mathematical Introduction* (2nd edition). Springer–Verlag.

Wolff, U. (1989). Collective Monte Carlo updating for spin systems. *Physical Review Letters,* **62,** 361–364.