J. R. Statist. Soc. B (2003)
65, *Part* 1, *pp.* 3–55

Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions

S. P. Brooks,

University of Cambridge, UK

P. Giudici

University of Pavia, Italy

and G. O. Roberts

Lancaster University, UK

[*Read before* The Royal Statistical Society *at a meeting organized by the* Research Section *at the 65th Annual Meeting of the Institute of Mathematical Statistics in Banff on Tuesday, July 30th, 2002*, Professor B. W. Silverman *in the Chair*]

Summary. The major implementational problem for reversible jump Markov chain Monte Carlo methods is that there is commonly no natural way to choose jump proposals since there is no Euclidean structure in the parameter space to guide our choice. We consider mechanisms for guiding the choice of proposal. The first group of methods is based on an analysis of acceptance probabilities for jumps. Essentially, these methods involve a Taylor series expansion of the acceptance probability around certain canonical jumps and turn out to have close connections to Langevin algorithms. The second group of methods generalizes the reversible jump algorithm by using the so-called saturated space approach. These allow the chain to retain some degree of memory so that, when proposing to move from a smaller to a larger model, information is borrowed from the last time that the reverse move was performed. The main motivation for this paper is that, in complex problems, the probability that the Markov chain moves between such spaces may be prohibitively small, as the probability mass can be very thinly spread across the space. Therefore, finding reasonable jump proposals becomes extremely important. We illustrate the procedure by using several examples of reversible jump Markov chain Monte Carlo applications including the analysis of autoregressive time series, graphical Gaussian modelling and mixture modelling.

Keywords: Autoregressive time series; Bayesian model selection; Graphical models; Langevin algorithms; Mixture modelling; Optimal scaling

1. Introduction

The reversible jump algorithm (Green, 1995) is an extension of the popular Metropolis–Hastings algorithm, designed to allow movement between different dimensional spaces. These algorithms are most commonly applied to (Bayesian) model determination problems (Dellaportas and Forster, 1999; Richardson and Green, 1997; Fan and Brooks, 2000) though other applications exist (e.g. Møller (1999) and Brooks *et al.* (2003)). We shall focus on the Bayesian model determination problem here and consider issues such as the choice of prior and specification of the likelihood as beyond the scope of the paper. Thus, we are concerned solely with using reversible

Address for correspondence: S. P. Brooks, Statistical Laboratory, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WB, UK.
E-mail: steve@statslab.cam.ac.uk

 1369–7412/03/65003

jump Markov chain Monte Carlo (MCMC) methods to obtain samples from some prespecified target distribution regardless of its derivation, although statistical considerations play an important role in the methods that we introduce.

In practice, the application of reversible jump methodology has predominantly remained within the domain of the MCMC expert, owing both to difficulties in constructing appropriate algorithms and to a common perception that it is particularly difficult to implement. So, though the scope for applications is vast, the full potential of this and similar methods will remain unrealized until fundamental implementation aspects have been solved. Perhaps the greatest of these is the problem of constructing proposed moves in complicated non-standard spaces, since there is no natural neighbourhood structure between models to guide us.

The aim of this paper is to provide both a general framework for constructing these jumps and for automating the process of choosing proposals efficiently. There are two main methodological ideas to our approach. Firstly, we introduce a collection of techniques that can be used to scale and shape proposal distributions automatically. Then, secondly, we extend reversible jump methodology to a more general auxiliary variable (AV) framework which can be used to improve Markov chain mixing properties by introducing temporary biases to assist the exploration of the discrete model space.

We begin by introducing the general reversible jump MCMC methodology both to establish the notation to be used throughout the paper and to motivate discussion of the potential implementational difficulties that are associated with the method.

1.1. Reversible jump Markov chain Monte Carlo methodology

In this subsection, we introduce reversible jump MCMC methodology in a fairly general setting. Though the introduction of models is not strictly necessary at this stage, we suppose (to keep in mind a motivating example in Bayesian inference) that we have models $M_1, \ldots, M_k, \ldots$, where model M_i has a continuous parameter space. The ideas that we develop can be easily extended to the partially or totally discrete cases but we restrict our attention to only the continuous case in this paper.

We write $\pi(M_i, \boldsymbol{\theta}_i)$ for the density part of our target distribution π restricted to M_i. Thus, for an arbitrary set B,

$$\pi(B) = \sum_i \int_{B \cap \Theta_i} \pi(M_i, \boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_i.$$

We denote the parameter space for M_i as Θ_i and, with a minor abuse of notation, we write $\boldsymbol{\theta}_i$ (a vector of length n_i) for a typical element of Θ_i. We focus on moves between M_i and M_j with $n_i < n_j$. By reversibility, this also characterizes the reverse move, and moves between all collections of pairs of models can be dealt with similarly.

Green (1995) provided the following general formulation for transdimensional jumps between $\boldsymbol{\theta}_i$ in model M_i and $\boldsymbol{\theta}_j$ in model M_j. Given that the chain is currently in state $(M_i, \boldsymbol{\theta}_i)$, we propose a new value for the chain $(M_j, \boldsymbol{\theta}_j)$ from some proposal distribution $Q(\boldsymbol{\theta}_i, \mathrm{d}\boldsymbol{\theta}_j)$, which is then subsequently either accepted or rejected. Green (1995) showed that, if $\pi(\mathrm{d}\boldsymbol{\theta}_i)\, Q(\boldsymbol{\theta}_i, \mathrm{d}\boldsymbol{\theta}_j)$ is dominated by a symmetric measure μ and has Radon–Nikodym derivative with respect to μ given by $R(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, then detailed balance is preserved if we accept the proposed new state with probability $\min\{1, A_{i,j}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)\}$, where

$$A_{i,j}([M_i, \boldsymbol{\theta}_i], [M_j, \boldsymbol{\theta}_j]) = \frac{R(\boldsymbol{\theta}_j, \boldsymbol{\theta}_i)}{R(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)}.$$

The general formulation can be simplified in the context of most model selection problems by restricting attention to certain jump constructions, as follows. To move from model M_i to M_j, we generate a random vector $\mathbf{V}$ of length $n_j - n_i$ consisting of variables drawn from some proposal density $\varphi(\cdot)$. We denote the joint density of $\mathbf{V}$ by

$$\varphi_{n_j-n_i}(\mathbf{v}) = \prod_{i=1}^{n_j-n_i} \varphi(v_i).$$

Having generated $\mathbf{V}$, we now propose the move from $\boldsymbol{\theta}_i$ to $\boldsymbol{\theta}_j = f_{i,j}(\boldsymbol{\theta}_i, \mathbf{V})$, where the so-called jump function $f_{i,j} : \Theta_i \times \mathbb{R}^{n_j-n_i} \to \Theta_j$ denotes an injection, mapping the current state of the chain together with the generated random vector to a point in the higher dimensional space. This move is then accepted with probability

$$\alpha\{(M_i, \boldsymbol{\theta}_i), (M_j, \boldsymbol{\theta}_j)\} = \min\{1, A_{i,j}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)\},$$

where $A_{i,j}$ takes the familiar form (Green, 1995)

$$A_{i,j}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \frac{\pi(M_j, \boldsymbol{\theta}_j)\, r_{ji}(\boldsymbol{\theta}_j)}{\pi(M_i, \boldsymbol{\theta}_i)\, r_{ij}(\boldsymbol{\theta}_i)\, \varphi_{n_j-n_i}(\mathbf{v})} \left| \frac{\partial f_{i,j}(\boldsymbol{\theta}_i, \mathbf{v})}{\partial(\boldsymbol{\theta}_i, \mathbf{v})} \right| \tag{1}$$

and $r_{ij}(\boldsymbol{\theta}_i)$ denotes the probability that a proposed jump to model j is attempted at any particular iteration, starting from $\boldsymbol{\theta}_i$ in Θ_i. For notational convenience, we shall refer to the final (Jacobian) term in equation (1) as $J_{ij}^{f}(\boldsymbol{\theta}_i, \mathbf{v})$.

In the case where $n_i > n_j$, we just take

$$A_{i,j}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = A_{j,i}(\boldsymbol{\theta}_j, \boldsymbol{\theta}_i)^{-1}.$$

In this paper, we shall focus mainly on this particular implementation of the general reversible jump algorithm, since many practical applications adopt this form, and therefore all our examples involve this special case. However, the methods and results provided here can be extended to the more general case and we shall describe these using the saturated space approach of Section 5 and Section 6.

Consider our motivating example where π is the posterior distribution over a collection of models $M_1, \ldots, M_k, \ldots,$ with prior model probabilities $p(M_1), \ldots, p(M_k), \ldots$ respectively, and within-model prior densities $\{p_i(\boldsymbol{\theta}_i),\ \boldsymbol{\theta}_i \in \Theta_i\}$. Assume that within each model M_i the likelihood is given by $L_i(\text{data}|\boldsymbol{\theta}_i)$. Then, the target density is the corresponding posterior density given by

$$\pi(M_i, \boldsymbol{\theta}_i) \propto L_i(\text{data}|\boldsymbol{\theta}_i)\, p_i(\boldsymbol{\theta}_i)\, p(M_i). \tag{2}$$

Many modifications, extensions and variations of reversible jump methodology exist. For instance, a convenient mathematical framework for describing the composite state space for all models is to use the state space of a suitable marked Poisson process. In this framework, the target distribution can be written as a density with respect to a chosen marked Poisson process measure. To move around such a space, it is natural to use birth-and-death processes (see for example Preston (1977) and Ripley (1977)) and this is the approach that is used to simulate interacting spatial point processes in Geyer and Møller (1994) and the considerable body of work that leads from this (see for example the review of Kendall and Thonnes (1999)). This approach was also used in Stephens (2000) and applied effectively to the problem of Bayesian inference for mixtures with an unknown number of components.

An approach to transmodel dynamics, in which the algorithm stores a vector for each model at every iteration, has been introduced by Carlin and Chib (1995) and recently extended to a

general framework, which also includes the reversible jump algorithm as described above, by Godsill (2001). As a direct extension of the reversible jump, Green and Mira (2001) introduced a procedure for reassigning rejected moves by sequentially attempting further proposals using a modification of the usual accept–reject mechanism. This procedure allows some level of adaptation in the sense that later proposals at any particular iteration can be allowed to use information gained from earlier rejections within that iteration. These and other related advances are reviewed in more detail in Green (2002).

All these extended methodologies offer alternative frameworks in which to construct Markov chain dynamics to explore model space. However, beyond that, there has been little progress in the problem of exactly how to construct proposal distributions. For algorithms on Euclidean spaces, the metric structure of the state space guides the construction of the proposal. For instance for the random walk Metropolis algorithm on a continuous target density, using a proposal distribution with variance σ^2, very small values of σ^2 will lead to small jumps which are almost all accepted, whereas large values will lead to an excessively high rejection probability for proposed moves. Thus, the *scaling* problem will typically have an optimal value for the proposal scale σ which lies between these two extremes. Although proposal distributions can be refined in more subtle ways than by variance alterations, the restriction of the choice of proposal problem to this one-dimensional problem is appealing, works well in practice and is supported in part by currently available theory (see for example Roberts *et al*. (1997)).

The problem for reversible jump moves is that there is no direct analogue of this kind of scaling problem, since there is no natural notion of a 'local' move with an arbitrary high acceptance probability. The approach of this paper is to translate natural ideas for the construction of proposals (such as the scaling problem) from their natural Euclidean environment to the union of model spaces.

1.2. Example: autoregressive model choice

To illustrate and motivate the methodology that we propose, we use a simple example of model choice for autoregressive time series models of unknown order. Here we describe a standard implementation of reversible jump methods to this problem (see for example Godsill (2001) and Ehlers and Brooks (2002)).

Suppose that we have data $x_1, \ldots, x_T$ from an autoregressive process of unknown order. Let model M_k correspond to the kth-order autoregressive process which is specified by the relational formula

$$X_t = \sum_{\tau=1}^{k} a_\tau X_{t-\tau} + \varepsilon_t, \qquad t = k_{\max} + 1, \ldots, T,$$

for $k = 1, 2, \ldots, k_{\max}$, where $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$. To model the data, we assume a uniform prior for k and within model M_k we take independent $N(0, \sigma_a^2)$ priors for the coefficients a_i, $i = 1, \ldots, k$ (thereby not imposing stationarity), and assume that σ_ε^2 takes an inverse gamma prior. In practice, we might wish to take a fairly large value for σ_a^2 to reflect a degree of prior uncertainty about the values that these parameters might take, though this may begin to affect the model probabilities if σ_a^2 is too large; see Berger (2000). We note that, if we had taken the normal–inverse χ^2 prior (Gelman *et al.* (1995), section 3.3) for the autoregressive parameters, then the posterior marginals and even the posterior model probabilities would be analytically tractable. However, for illustration, we shall adopt the priors described above and, since the issue of the specification of the prior is not of direct relevance here, we shall not discuss such issues further in this paper. We assume only that a model and prior have been prescribed and that we wish to implement an efficient reversible jump MCMC sampler to explore the corresponding posterior.

The number of autoregressive parameters in model M_k is given by $n_k = k$ and we can write $\boldsymbol{\theta}_k = (a_1, a_2, \ldots, a_k)$, dropping the σ_ε^2-parameter which will be present in every model. The posterior therefore comprises three terms: the likelihood

$$L_k(\mathbf{x}|\boldsymbol{\theta}) = \prod_{t=k_{\max}+1}^{T} \frac{1}{\sqrt{(2\pi\sigma_\varepsilon^2)}} \exp\left\{ -\frac{1}{2}\left(\frac{x_t - \sum_{\tau=1}^{k} a_\tau x_{t-\tau}}{\sigma_\varepsilon} \right)^2 \right\},$$

the multivariate normal prior for the autoregressive parameters and the uniform prior over model order; $k = 1, \ldots, k_{\max}$. Note that we are using an approximate likelihood form here (Ehlers and Brooks, 2002) and conditioning on the initial $k_{\max}$ observations under all models.

In this particular example, jumps take place only between nested models differing in dimension by 1 at most. Because of the nesting structure, a natural function $f_{k,k+1}$ linking the two parameter spaces is the identity, so

$$f_{k,k+1}(\boldsymbol{\theta}_k, v) = (\boldsymbol{\theta}_k, v),$$

i.e. we set the new parameter in the larger model $a_{k+1} = v$. In this case, the determinant term in equation (1) is simply 1.

Suppose that at any iteration we propose a jump from $\boldsymbol{\theta}_k$ in M_k to $\boldsymbol{\theta}_{k+1} = (\boldsymbol{\theta}_k, v)$ in M_{k+1} with probability $r_{k,k+1}$ (independently of $\boldsymbol{\theta}_k$ and with the reverse move having probability $r_{k+1,k}$); otherwise we propose a jump which decreases the dimension of the model by 1. Here, we might simply choose to increase or decrease the model order with equal probability except at the extremes, so that $r_{k,k+1} = r_{k,k-1} = \frac{1}{2}$ if $0 < k < k_{\max}$, and $r_{0,1} = r_{k_{\max},k_{\max}-1} = 1$.

If we propose to increase the dimension, we might take as our proposal $v \sim N(0, \sigma^2)$ (we shall look at more general proposals later) and applying expression (2) to equation (1) we obtain

$$A_{k,k+1}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{k+1}) = \frac{L_{k+1}(\mathbf{x}|\boldsymbol{\theta}_k, v)}{L_k(\mathbf{x}|\boldsymbol{\theta}_k)} \frac{p_{k+1}(\boldsymbol{\theta}_{k+1})}{p_k(\boldsymbol{\theta}_k)} \frac{r_{k+1,k}}{r_{k,k+1}} \frac{1}{\varphi(v)}, \tag{3}$$

where $\varphi(v) = (2\pi\sigma)^{-1/2} \exp(-v^2/2\sigma^2)$. Since we have independent priors for the model parameters, the prior term in the denominator will cancel with the corresponding terms in the numerator. These reversible jump moves will be interspersed with suitably chosen within-model Metropolis–Hastings moves to explore the posterior.

The overall performance of the resulting algorithm will depend on our choice of proposal parameters. In particular, the ability to mix between models will depend heavily on our choice for σ^2. If the proposal variance is too small, jumps to simpler models will be rare, since the φ-term (which will lie in the numerator for this move) will be small. By reversibility, the algorithm will show an equal reluctance to perform the reverse move also. Similarly, if σ^2 is too large, moves increasing the dimensionality of the model will be rare, since the algorithm will propose many values far from areas of high posterior support. In practice, σ^2 is usually 'tuned' on the basis of short pilot runs. However, for more complex problems, this can be (sometimes impractically) difficult and time consuming.

1.3. General perspective

The reversible jump MCMC technique is a very general and widely applicable technique. Markov chain convergence is assured under very weak conditions on the jump function $f_{i,j}$ and the proposal density φ. However, very little is known about how to do this efficiently in

a generic way. Although there has been considerable progress in this area for fixed dimension sampling problems (see for example Gelman *et al.* (1996) and Roberts and Rosenthal (1998)), most of the available statistical applications of reversible jump techniques rely on various strategies of empirical tuning, as discussed above. In this paper we discuss a variety of general recipes for choosing the jump function $f_{i,j}$ and for automatically scaling the proposal distribution φ.

There are two aspects to choosing successful algorithms: knowing roughly where to jump to and choosing jump distribution parameters appropriately. We shall discuss the two issues separately, bringing them together for our examples. Section 2 considers the first (qualitative) problem, whereas the later sections are all concerned with quantitative issues.

In Section 2, we develop a framework which allows us to discuss current and new methods for constructing reversible jumps. In particular, we consider the concept of moment matching (Green, 1995), and we introduce the idea of 'centring' reversible jumps on moves which, for example, may take advantage of what we call *weak non-identifiability*. We also introduce the notion of conditional maximization as a device for targeting jumps to higher dimensional models into appropriate regions.

One of the main contributions of this paper is the introduction of an automatic method for determining proposal parameters, based on an analysis of acceptance probabilities for these jumps. The ideas of locating and scaling proposal distributions is described in detail, and all the ideas are illustrated by using the simple autoregressive model choice example introduced above. In Section 4 we extend this approach to consider more general analyses of the acceptance probabilities for determining automatic scales and we provide some technical discussion on the advantages of adopting these higher order approaches.

Throughout the paper, we make connections to more familiar algorithms on Euclidean spaces, such as the random walk Metropolis and Langevin algorithms. This is because we can describe such algorithms in terms of conditions on the acceptance probability formula, and these notions can be easily transported to the reversible jump context.

In Section 5 we provide a generalization of the reversible jump algorithm by using the so-called saturated space approach. This method allows the random seeds generated to increase the dimensionality of the current model to be retained on the subsequent simplification of the model. In Section 6, through artificially introduced dependences between these random seeds, algorithms with a kind of *momentum* through model space can be generated, potentially aiding mixing in hard problems with multimodality in the model space. We provide examples of the implementation of these methods and illustrate the dramatic improvement in performance that we observe in the context of several real applications.

There are three distinct settings which together describe the vast majority of current applications of reversible jump MCMC methods: variable selection, where we decide which parameters should be included in a model, association selection, where we decide which interactions exist between a fixed number of model parameters, and finally classification problems, where we decide how to assign observations to different groups within a model. The implementational details for our autoregressive example (as an illustration of a variable selection problem) are discussed throughout the text and in Section 7.1 we report the results of the application of these algorithms in the context of a particular data set. To illustrate the applicability and utility of these methods further we also introduce more applications. In Section 8 we introduce an example of association selection in the context of graphical Gaussian modelling, and in Section 9 we introduce a classification problem by re-examining the mixture modelling problem of Richardson and Green (1997), showing how the performance of the sampler can be improved through the adoption of our proposed new methods.

2. Centring and transforming proposals

Reversible jump algorithms offer no real mathematical generalization over traditional fixed dimensional Metropolis–Hastings methods; see for example Tierney (1998). However, in a sense they are intrinsically more complex than methods on continuous distributions on $\mathbb{R}^k$.

To see this, consider the case where π is a continuous density on $\mathbb{R}^k$. The *vanilla* (or default) algorithm that we might use to explore π is the random walk Metropolis algorithm, perhaps proposing a move from $\mathbf{x}$ to $\mathbf{x}+\mathbf{V}$ where the $V_i \sim N(0, \zeta^2)$ say. The analogue of the acceptance ratio (1) in this context is simply

$$A(\mathbf{x}, \mathbf{x}+\mathbf{V}) = \frac{\pi(\mathbf{x}+\mathbf{V})}{\pi(\mathbf{x})}.$$

Here A takes the value 1 at a central move corresponding to all of the V_i taking the value 0 and therefore proposing to remain at $\mathbf{x}$. By continuity therefore, small jumps (i.e. where $|\mathbf{V}|$ is small) are accepted with a very high probability and, importantly, so is the reverse move. Thus, depending on π, a sufficiently small ζ^2 defines an algorithm with a sufficiently high acceptance rate to allow the chain to move around the state space (though sometimes perhaps rather slowly). See Gelman *et al.* (1996) and Roberts *et al.* (1997) for results describing the relationship between the optimal choice of ζ^2 (in terms of the corresponding convergence rate of the chain) and the acceptance rate. Although none of these absolutely guarantees a useful algorithm in practice, this all makes the tuning of algorithms (choosing ζ^2) relatively straightforward. The existence of a central move where $A = 1$ is certainly not guaranteed in the reversible jump framework, and the aims of the simplest algorithms that we introduce will be to ensure that it does.

More sophisticated algorithms for target densities on $\mathbb{R}^k$ can improve the efficiency over random walk Metropolis algorithms drastically. So-called Langevin algorithms (see for example Roberts and Tweedie (1996)) use gradient information about π to propose candidate moves which are more likely to be accepted. Specifically, given $\mathbf{x}$ the algorithm proposes a move to $\mathbf{x}+\mathbf{V}$ where

$$\mathbf{V} \sim N\left[0 + \frac{\zeta^2}{2}\nabla \log\{\pi(\mathbf{x})\}\zeta^2\right].$$

The usual way to motivate this choice of proposal is to consider the discretization of a suitable *Langevin diffusion* which has π as its stationary distribution. In the context of choosing sensible proposals, we note that this proposal leads to an acceptance ratio A which satisfies $A(\mathbf{x}, \mathbf{x}) = 1$ and

$$\left.\frac{\partial A(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}}\right|_{\mathbf{y}=\mathbf{x}} = 0. \tag{4}$$

Intuitively, this allows A to be close to 1 further away from $\mathbf{x}$ than for the random walk Metropolis case, therefore allowing more ambitious jumps to be proposed and accepted. This leads to improved mixing (see Roberts and Rosenthal (1998) for theoretical results on this).

In the reversible jump context, we shall show that it is possible to construct algorithms which mimic the behaviour of these fixed dimensional examples in terms of the characterization given in equation (4). In fact even higher order Langevin algorithms have analogues in the reversible jump context and in Section 4 we demonstrate how these can be constructed.

2.1. Constructing the jump function

We return to the general setting of Section 1.1. Within this framework, the jump function $f_{i,j}$ can be chosen arbitrarily (subject only to certain differentiability properties that we shall need for the methods that we introduce) though often some advantage is obtained by exploiting features such as nesting and moment matching between models.

2.1.1. Example: nested jumps

Suppose that $\boldsymbol{\theta}_i = (\mu_1, \ldots, \mu_{n_i})$ for a collection of parameters μ_i whose interpretation is either unaltered in models of increased complexity or at least whose values are unlikely to change much in moving from a smaller to a larger model. Then one natural set of constraints, when proposing a move to a more complex model, preserves the values of all current parameters.

2.1.2. Example: moment matching

Suppose that $\boldsymbol{\theta}_i = (\mu_1^i, \ldots, \mu_{n_i}^i)$ and $\boldsymbol{\theta}_j = (\mu_1^j, \ldots, \mu_{n_j}^j)$. $f_{i,j}$ is often chosen with the aid of an $(n_j - n_i)$-dimensional constraint, often on the basis of moment matching requirements (Green, 1995). For example suppose that B is an $n_i \times n_j$ matrix; then we might wish to impose the condition that $B\boldsymbol{\theta}_j = \boldsymbol{\theta}_i$. The simplest example of this occurs where $n_j = n_i + 1$ and the dimension matching constraints are $\mu_k^i = \mu_k^j$, $k = 1, 2, \ldots, n_i - 1$, and $(\mu_{n_i}^i + \mu_{n_i+1}^j)/2 = \mu_{n_i}^i$. A corresponding jump function might then be to set $f_{i,j}(\boldsymbol{\theta}_i, v) = (\mu_1^i, \ldots, \mu_{n_i-1}^i, \mu_{n_i}^i - v, \mu_{n_i}^i + v)$. In this way the first moment of the two new parameters matches that of the single parameter that they replace.

As it happens, even for a given collection of dimension matching constraints, there is still some flexibility in how to choose the jump function. We assume here that we are given some collection of *canonical jump functions* $\{f_{i,j}\}$, $f_{i,j} : \Theta_i \times \mathbb{R}^{n_j - n_i} \to \Theta_j$. For instance, in the nested case above, it is natural to take $f_{i,j}$ to be the identity function. The proposed new state is then given by

$$\boldsymbol{\theta}_j = h_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}) = f_{i,j}\left\{\boldsymbol{\theta}_i, v_{i,j,\boldsymbol{\theta}_i}(\mathbf{u})\right\} \tag{5}$$

where $v_{i,j,\boldsymbol{\theta}_i}$ plays the role of a general *proposal transformation* of some canonical random variable (seed) $\mathbf{u}$. We typically choose $v_{i,j,\boldsymbol{\theta}_i}$ from a low dimensional family of functions. All that we insist on is that $v_{i,j,\boldsymbol{\theta}_i}$ is invertible.

The function $f_{i,j}$ will be considered fixed in our search for an automatic proposal mechanism. However, we still retain considerable freedom for the choice of $v_{i,j,\boldsymbol{\theta}_i}$. This, and the following section, will largely concentrate on strategies for the automatic choice of $v_{i,j,\boldsymbol{\theta}_i}$ which can be adopted without additional computation and complexity inherent in the pilot tuning required for many existing reversible jump applications.

Thus, in this general set-up, if q denotes a state-independent proposal for the canonical seeds $\mathbf{u}$, then the acceptance ratio given in equation (1) now becomes

$$A_{i,j}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \frac{\pi(M_j, \boldsymbol{\theta}_j)\, r_{ji}(\boldsymbol{\theta}_j)}{\pi(M_i, \boldsymbol{\theta}_i)\, r_{ij}(\boldsymbol{\theta}_i)\, q_{n_j - n_i}(\mathbf{u})} \left| \frac{\partial h_{i,j}(\boldsymbol{\theta}_i, \mathbf{u})}{\partial(\boldsymbol{\theta}_i, \mathbf{u})} \right|. \tag{6}$$

Once again, we shall refer to the final (Jacobian) term in equation (6) as $J_{ij}^h(\boldsymbol{\theta}_i, \mathbf{u})$ and, for notational convenience, drop the i-, j- and $\boldsymbol{\theta}_i$-subscripts on the majority of terms for the remainder of the paper, since we shall always consider the case in which we move from state $\boldsymbol{\theta}_i$ in model M_i to some new state in model M_j.

2.2. Centring proposals

To extend our mechanism for constructing reversible jump algorithms beyond the simple nested case, we require the introduction of a collection of centring functions. A *centring* function $c_{i,j} : \Theta_i \to \Theta_j$ can be specified by the equation

$$c(\boldsymbol{\theta}_i) = f[\boldsymbol{\theta}_i, v\{b(\boldsymbol{\theta}_i)\}]$$

(dropping the subscripts on c), where $b(\boldsymbol{\theta}_i) \equiv b_{i,j}(\boldsymbol{\theta}_i)$ is some real-valued function, often taken to be identically zero. Essentially, we are identifying a 'special' value $b(\boldsymbol{\theta})$ for the proposal vector $\mathbf{u}$ and associating this with a point c in the higher dimensional space. We shall need such a function between each collection of models for which we might attempt to jump. Intuitively, the centring function should choose a 'representative point' on the image of $h(\boldsymbol{\theta}_i, \mathbf{u})$ from which to extract information for the construction of v.

There are various possible strategies for constructing the centring functions by using statistical or mathematical principles and we discuss some alternatives in Section 2.4. However, for the moment we shall focus on a centring function defined according to so-called weak non-identifiability, under which the probability model described by $\boldsymbol{\theta}_i$ in Θ_i is identical with that described by $c(\boldsymbol{\theta}_i)$ in Θ_j. As an illustration, in the autoregressive example of Section 1.2, the *weak non-identifiability centring* for a move between M_k and M_{k+1} is characterized by finding a point b such that $v(b) = 0$ and so $c(\boldsymbol{\theta}_k) = (\boldsymbol{\theta}_k, 0)$, since the k-dimensional model with parameters $(a_1, \ldots, a_k)$ is identical (in terms of likelihood contribution) with the $(k+1)$-dimensional model with parameters $(a_1, \ldots, a_k, 0)$. Thus, it would be natural to centre any proposal for the parameter vector in the $(k+1)$-dimensional model about this point, which corresponds to the k-dimensional model in the higher dimensional space.

To choose $\{v_{i,j,\boldsymbol{\theta}_i}\}$ for any particular problem, we propose a hierarchy of strategies in terms of both ease of implementation and accuracy (ease decreases with increasing accuracy). Before we discuss these strategies in detail, we note the alternative formulation of the approach described above which we describe in Appendix A and which eliminates the need to use both f and h at the expense of having more general state-dependent proposal densities. The notation that we adopt here is most appropriate for the scaling (and location) problems that we shall consider, because it distinguishes the transformation problem (that of modifying v) from the more general problem of choosing a link function between different dimensional spaces (that of choosing f). Fig. 1 summarizes the notation that we shall use throughout the paper. Fig. 1(a) shows the random variate $u \sim q$ and the point $b(\boldsymbol{\theta}_i)$ for a particular $\boldsymbol{\theta}_i$. These points are both mapped via the v-

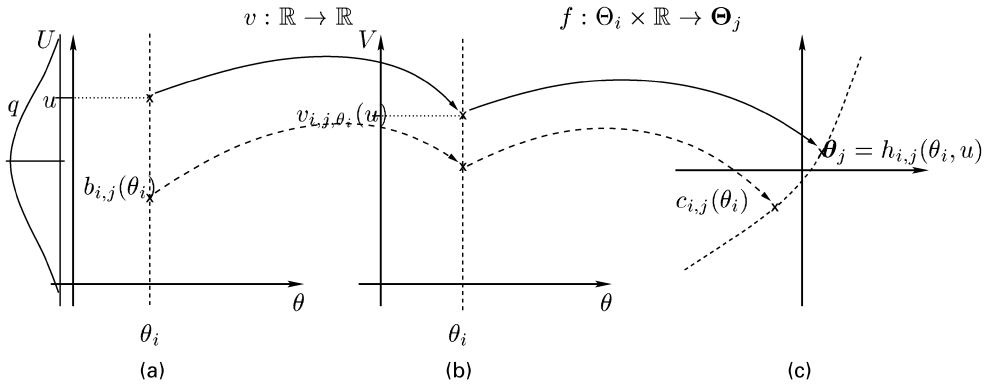

Fig. 1. Illustration of the notation that is used in the paper, demonstrating the use of the functions v and f to scale u and to move from $\Theta_i \times \mathbb{R}$ to Θ_j for $i = 1$ and $j = 2$ respectively

function onto Fig. 1(b). Clearly this mapping depends on the current state $\boldsymbol{\theta}_i$. $v(u)$ and $v(b)$ are then both mapped to points in Θ_j. $v(b)$ becomes mapped to the centring point $c \in \Theta_j$, whereas $v(u)$ becomes mapped to some point $h(\boldsymbol{\theta}_i, u)$. Different values of u would give rise to different points along the broken line through c with a value of $u = b$ mapping to the centring point. Thus, the density q essentially becomes mapped to a density along the broken line through b in Θ_j.

2.3. Example: the autoregressive example revisited

Suppose that we wish to move from $\boldsymbol{\theta}_k$ to $\boldsymbol{\theta}_{k+1}$ by generating a random value $u \sim N(0, 1)$ independently of the state of the chain. Since the models here are nested, a natural choice is to adopt the identity as the jump function f and to take a simple linear proposal transformation $v(u) = \sigma u$. This then provides the same jump transition as described in Section 1.2. σ may depend on k and/or $\boldsymbol{\theta}_k$, but we shall drop the notational dependence here for clarity.

Using weak non-identifiability centring, we choose $b(\boldsymbol{\theta}_k) = 0$, since $L_{k+1}\{\mathbf{x}|(\boldsymbol{\theta}_k, 0)\} = L_k(\mathbf{x}|\boldsymbol{\theta}_k)$ and the centring function is simply $c(\boldsymbol{\theta}_k) = (\boldsymbol{\theta}_k, 0)$. In this case, the Jacobian term in equation (6) is simply σ, and we obtain the acceptance probability in equation (3) by substituting into equation (6).

2.4. The conditional maximization approach

Other centring methods have recently been proposed in specific contexts. See for example Green (2002) for results based on a Gaussian approximation and Ntzoufras *et al.* (2002) for applications to generalized linear models. Here, we introduce a general scheme which may be used to augment any of the methods discussed in previous sections, removing an additional degree of freedom by finding a sensible location for the proposal distribution. The *conditional maximization* scheme improves on the non-identifiability centring scheme in that, whereas the non-identifiability centring method is restricted to jumps between nested models, at least in theory, the conditional maximization method can be applied in any setting.

The idea behind all our methods is to find proposal parameters that improve the chances that the chain actually jumps from one model to the next. When considering where to jump to in the higher dimensional space, an obvious place to locate the proposal is around posterior modes in the higher dimensional model. The conditional maximization scheme proceeds by maximizing the posterior distribution $\pi\{M_j, h(\boldsymbol{\theta}_i, \mathbf{u})\}$ with respect to $\mathbf{u}$ to obtain the maximizing value $\hat{\mathbf{u}}$, say. Then our centring point is chosen so that $c(\boldsymbol{\theta}_i) = h(\boldsymbol{\theta}_i, \hat{\mathbf{u}})$. Thus, we are essentially conditioning on the current state $\boldsymbol{\theta}_i$ and centring at the posterior conditional mode. In practice, the value of $\hat{\mathbf{u}}$ is obtained by setting the derivative (with respect to $\mathbf{u}$) of the log-posterior distribution under the higher model to 0. To derive the remaining proposal parameters (typically, the scale at least) we can use the scaling methods that are described in the next sections, centring at the posterior mode.

The conditional maximization method provides an alternative to the weak non-identifiability centring described above and is not restricted to the case where jumps occur only between nested models. As an illustration, if we return to our autoregressive model, and we wish to take $u \sim N(\mu, \sigma^2)$, then using conditional maximization we would set μ to be the value of u maximizing $L_{k+1}(\mathbf{x}|\boldsymbol{\theta}_i, u)\, p_{k+1}(\boldsymbol{\theta}_i, u)$, obtaining

$$\mu = \frac{\displaystyle\sum_{t=k_{\max}+1}^{T} \left(x_t - \sum_{\tau=1}^{k} a_\tau x_{t-\tau} \right) x_{t-k-1}}{\displaystyle\sum_{t=k_{\max}+1}^{T} x_{t-k-1}^2 + \sigma_\varepsilon^2/\sigma_a^2}. \tag{7}$$

3. Automatic proposal scaling: the zeroth-order method

Here we shall introduce a simple and easy-to-implement method for automatically choosing proposal scales. Suppose that we are currently in state $\boldsymbol{\theta}_i$; then we wish to choose a scale for the proposal transformation v that can be used to generate a state in the new model, $\boldsymbol{\theta}_j$. We choose the scale so that, for the jump between $\boldsymbol{\theta}_i$ and its image in Θ_j under the centring function $c(\boldsymbol{\theta}_i)$, the acceptance ratio given in equation (6) is identically equal to 1, i.e.

$$A\{\boldsymbol{\theta}_i, c(\boldsymbol{\theta}_i)\} = 1, \tag{8}$$

More specifically, suppose that we constrain v to belong to the scaling family

$$v(\mathbf{u}) = \sigma \times \mathbf{u}, \tag{9}$$

where $\sigma \equiv \sigma_{i,j,\boldsymbol{\theta}_i}$ is a state-dependent scale parameter which may or may not be a standard deviation in general. Then, rearranging equation (8) using the definitions of A in equation (6) and v in equation (9), and by noting that

$$|J^h| = |J^f| \left| \frac{\partial \mathbf{v}}{\partial \mathbf{u}} \right|,$$

we obtain

$$\sigma^{n_j - n_i} = \frac{\pi(M_i, \boldsymbol{\theta}_i)\, r_{ij}(\boldsymbol{\theta}_i)\, q\{b(\boldsymbol{\theta}_i)\}}{\pi\{M_j, c_{i,j}(\boldsymbol{\theta}_i)\}\, r_{ji}\{c(\boldsymbol{\theta}_i)\}} \frac{1}{|J^f(\boldsymbol{\theta}_i)|} \tag{10}$$

giving a solution $\sigma\ (= \sigma(\boldsymbol{\theta}_i))$. Here we use the abbreviation $J^f(\boldsymbol{\theta}_i)$ to denote the value of $J^f(\boldsymbol{\theta}_i, \mathbf{v})$ at $v = v\{b(\boldsymbol{\theta}_i)\}$.

Setting $A(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = 1$ for certain 'central' jumps is normally a sound heuristic principle. For general MCMC algorithms, it is often automatically satisfied. For instance, in Euclidean spaces the random walk Metropolis algorithm on a continuous density will have A close to 1 for all sufficiently small jumps. Moreover, more sophisticated algorithms (for instance the hybrid Monte Carlo procedure of Duane *et al.* (1987)) can often be motivated in terms of arguments that try to fix A to be equal to, or approximately equal to, 1, for appropriately chosen jumps. For reversible jump algorithms, the lack of Euclidean structure in the state space means that obtaining $A(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = 1$ for appropriate jumps is not automatic. The zeroth-order method described above ensures that the acceptance probability does equal 1 for centred jumps between $\boldsymbol{\theta}_i$ and $c(\boldsymbol{\theta}_i)$. Further discussion of the ideas motivating the zeroth-order method appears in Section 3.2.

3.1. The zeroth-order method for Bayesian applications

To apply our procedure, we need to specify the functions f and $c(\boldsymbol{\theta}_i)$ for all collections of models M_i and M_j between which jumps might conceivably be proposed.

In the important situation where the distribution to be sampled has a density which can be decomposed as the likelihood times the prior, and where we have the option of using weak non-identifiability centring, a useful strategy for specifying $c(\cdot)$ proceeds in the following way. Suppose that there exists $b(\boldsymbol{\theta}_i)$ such that $L_i(\text{data}|\boldsymbol{\theta}_i) = L_j[\text{data}|f\{\boldsymbol{\theta}_i, b(\boldsymbol{\theta}_i)\}]$; then set $c(\boldsymbol{\theta}) = f\{\boldsymbol{\theta}_i, b(\boldsymbol{\theta}_i)\}$. A location shift in f ensures that there is no loss of generality in taking $b(\boldsymbol{\theta}_i) = 0$ as the weak non-identifiability centring point.

Using weak non-identifiability centring, equation (10) reduces to

$$R^{n_j - n_i} = \frac{p_i(\boldsymbol{\theta}_i)\, p(M_i)\, r_{ij}(\boldsymbol{\theta}_i)\, q\{b(\boldsymbol{\theta}_i)\}}{p(M_j)\, p_j\{c(\boldsymbol{\theta}_i)\}\, r_{ji}\{c(\boldsymbol{\theta}_i)\}} \frac{1}{|J^f(\boldsymbol{\theta})|} \tag{11}$$

which is independent of the likelihood entirely. Of course this approach can be generalized to the situation where alternative factorizations of the target density are possible and centring

can be performed in any manner that allows a particular term to cancel. Beyond the statistical context there is then no particular reason to think that the centring point that is produced by this method is a sensible location about which to construct the proposal distribution. However, within the statistical context with non-identifiable centring, this choice of centring function is entirely natural.

The fact that the likelihood drops out of equation (11) provides an important computational advantage in situations where the likelihood is expensive to compute, but it means that the proposal is being tailored to the prior rather than the posterior. When the prior and posterior are similar (as in the graphical Gaussian model example described later), the zeroth-order method provides a very simple and efficient proposal-generating mechanism. However, the zeroth-order method may perform poorly when the prior and posterior differ greatly. In such cases, the method may be improved if we can also incorporate information from the data in choosing the proposal scales. A natural way to do this is to consider higher order approximations, as we shall see in Section 4.

3.1.1. Example: the autoregressive example revisited

Applying weak non-identifiability centring and the zeroth-order algorithm to the autoregressive example introduced in Section 1.2, and adopting the identity jump function f with simple linear scale function $v(u) = \sigma u$, the acceptance ratio at the centring point, $(\boldsymbol{\theta}_k, 0)$ in equation (6) reduces to

$$A\{\boldsymbol{\theta}_k, (\boldsymbol{\theta}_k, 0)\} = \frac{p_{k+1}(\boldsymbol{\theta}_k, 0)}{p_k(\boldsymbol{\theta}_k)} \frac{p(M_{k+1})}{p(M_k)} \frac{r_{k+1,k}}{r_{k,k+1}} \frac{\sigma}{q(0)}. \tag{12}$$

Setting the acceptance ratio to 1, equation (12) becomes

$$A\{\boldsymbol{\theta}_k, (\boldsymbol{\theta}_k, 0)\} = \frac{1}{(2\pi\sigma_a^2)^{1/2}} \frac{r_{k+1,k}}{r_{k,k+1}} \frac{\sigma}{(2\pi)^{-1/2}} = 1, \tag{13}$$

which can be solved to obtain

$$\sigma^2 = \sigma_a^2 \left(\frac{r_{k,k+1}}{r_{k+1,k}} \right)^2. \tag{14}$$

As pointed out above, the resulting proposal variance is independent of the data and so only information from the prior is used to tune the proposal distribution in this case.

3.2. In support of the zeroth-order method

The acceptance probability of any proposed move is a non-decreasing function of A, so it might be tempting to think that, the higher A is, the better the algorithm's prospects of traversing the state space effectively. However, if A is large, then the reciprocal of A will be small and the reverse move becomes unlikely. Thus, by setting $A = 1$ we simultaneously maximize the probability of both the forward and the backward moves. The following example illustrates this further.

Suppose that we have two disjoint spaces, Θ_1 and Θ_2, with target probabilities p and $1 - p$ respectively. If Θ_1 and Θ_2 were just single-point spaces, the optimal Markov chain (in the sense of minimizing the transition matrix's second eigenvalue) which mixes throughout the space has transition matrix

$$P = \begin{pmatrix} 1 - \min\{1, (1-p)/p\} & \min\{1, (1-p)/p\} \\ \min\{1, p/(1-p)\} & 1 - \min\{1, p/(1-p)\} \end{pmatrix}, \tag{15}$$

i.e. this transition matrix minimizes the eigenvalue over all transition matrices with $(p, 1-p)$ as the stationary distribution. Either the transition from Θ_1 to Θ_2 or its reverse (or both) has probability 1. To proceed further we need to impose a little more structure on the state space.

Now, suppose that $\Theta = \Theta_1 \cup \Theta_2$ with $\Theta_1 = \{e\}$ and $\Theta_2 = [0, 1]$. Let $\pi(e) = p$, and $\pi(u) = (1-p), 0 \leqslant u \leqslant 1$. Consider the algorithm which attempts to jump from model i to model $3-i$ (alternating with within-model moves which sample independently from the distribution constrained within that model). Let the proposal density be $U(0, 1)$ if we are currently in M_1 attempting a move to the one-dimensional space M_2, and all moves from M_2 just attempt to jump to e. Now for the move from Θ_1 to Θ_2 we propose a uniform candidate on the interval $(\frac{1}{2} - R/2, \frac{1}{2} + R/2)$ and from equation (10) the zeroth-order method is selected by choosing $R = p/(1-p)$. If $p > \frac{1}{2}$, this occasionally selects moves outside the support of Θ_2—in fact this happens with probability $1 - (1-p)/p$. In this case the reverse move is always accepted. Conversely, if $p < \frac{1}{2}$, then all moves from θ_1 to Θ_2 are accepted, whereas reverse moves are only accepted from u-values within $(\frac{1}{2} - p/(2-2p), \frac{1}{2} + p/(2-2p))$. In either case, the process describing the current model state is in fact Markov and has the optimal transition probabilities described by equation (15).

Therefore, in adopting the zeroth-order method, we obtain the best possible mixing between models. (Note that this optimality is also related to the optimality of the Metropolis–Hastings rule among all accept–reject procedures; see Peskun (1973).) Of course if we knew *p a priori* (in the case $p > \frac{1}{2}$ for instance) we could design a different sampler which only attempted to move from Θ_1 to Θ_2 with probability $(1-p)/p$. Such a scheme would also achieve the optimal transition probabilities between models described in equation (15), and this would avoid the need to propose ridiculous moves, and therefore leading to some computational savings. However, in general, we shall not know p so such a strategy is not practically implementable. In this example it turns out that R for the zeroth-order method is in fact a function of p, but this is calculated directly from the probability of the jump between centring points, and not on the basis of the probability mass contained in each model.

Practical examples will never be as clear cut as this. It will be very rare that the model indicator itself will be a Markov chain for instance. However, the example illustrates what the zeroth-order method is attempting to achieve.

4. Extending the method

The zeroth-order method can be naturally extended by considering higher order expansion terms. The idea of trying to obtain A as close to 1 as possible can be used to motivate more sophisticated choices of proposal. One obvious way to extend the method is not only to fix A to take the value 1 at some chosen central value but also to stipulate that some of its derivatives be 0 at that central value also, so that A remains close to 1 within a region around this point. We begin by extending the zeroth-order scheme described in the previous section.

4.1. The first-order method

We now describe the simplest possible extension to the zeroth-order method, which we call the first-order algorithm. With A defined as in equation (6), this method satisfies both equation (8) and

$$\nabla A\{\boldsymbol{\theta}_i, c(\boldsymbol{\theta}_i)\} = \mathbf{0} \tag{16}$$

for all possible choices of i, j and $\boldsymbol{\theta}_i$. Here ∇ is taken with respect to $\mathbf{u}$, and therefore equation (16) imposes an $(n_j - n_i)$-dimensional constraint on the proposal. In practice, it is often easier to specify the derivative constraints in terms of the logarithm of the acceptance ratio.

This method can be thought of as the reversible jump analogue of the Langevin algorithm which is characterized by an equation similar to equation (16). Langevin algorithms tend to have considerably superior convergence properties than simpler zeroth-order methods (see for example Roberts and Rosenthal (1998)). This is because the algorithm takes into account local fluctuations in the shape of the target density and adjusts the target as a result. In addition, since the acceptance ratio is 1 except for a quadratic error (as opposed to linear in the zeroth-order case), larger jumps can be attempted without leading to acceptance rates close to 0.

As with the zeroth-order algorithm, we fix f and try to find a proposal density to satisfy both equation (8) and equation (16) simultaneously. Of course there are many different ways to do that. We give the following approach as an example.

4.1.1. *Example: first-order Gaussian proposals*

Suppose that $n_j - n_i = 1$. Then equations (8) and (16) together introduce a two-dimensional constraint on the proposal density. To satisfy these two conditions we need to consider only a class of distributions with 2 degrees of freedom. There are many possible choices for this, but we illustrate the idea with one of the most natural choices.

Suppose that we take u to be standard Gaussian and define v to be the linear function

$$v(u) = \mu + \sigma u.$$

Then, as long as the various density terms have analytically available first derivatives, equations (8) and (16) have easily available solutions. Note that solving $\nabla \log[A\{\boldsymbol{\theta}_i, c(\boldsymbol{\theta}_i)\}] = \mathbf{0}$ is usually easier than directly working with equation (16). A detailed worked example is given in Section 4.3, and a second generic family of first-order methods, using triangular shape proposals, is described in Appendix A.

4.2. *Higher order methods*

An extension to higher order methods need not stop at the first derivative. We may also set higher order derivatives to 0. Broadly speaking (at least for suitably differentiable target densities), as we set increasingly more derivatives to 0 we obtain acceptance probabilities which become increasingly closer to 1, at least in some neighbourhood of the centring point. However, usually additional computational costs are associated with the implementation of these higher order methods.

In practice, our proposal density will typically have only a few parameters which need to be selected. Given a proposal with r parameters we only need r constraints to specify those parameters. If we add additional constraints, then it may not be possible to solve all of them. As we shall see in the autoregressive example in Section 4.3, given a proposal with two parameters, these parameters may be set by taking the zeroth- and first-order constraints or by taking the first- and second-order constraints for example. In fact any combination of two constraints could be used and there is evidence to suggest that the flatness of the acceptance ratio is perhaps more important than its being closer to 1. We return to this point in greater detail in Section 4.4. In practice the choice of constraints may depend on analytic tractability and/or computational complexity.

4.3. Example: the autoregressive example revisited

We begin with the basic first-order method described by the simultaneous solution of equations (8) and (16) to obtain both a location and scale for the proposal transformation $v(u) = \mu + \sigma u$, i.e. $v = v(u) \sim N(\mu, \sigma^2)$. Setting $A = 1$ and the derivative of $\log(A)$ to 0 at the non-identifiability centring point $u = 0$, we obtain

$$\frac{\mu}{\sigma^2} = \frac{1}{\sigma_\varepsilon^2} \sum \left(x_t - \sum_{\tau=1}^{k} a_\tau x_{t-\tau} \right) x_{t-k-1}$$

and

$$\frac{1}{\sigma_a} r_{k+1,k} = \frac{1}{\sigma} \exp\left(-\frac{\mu^2}{2\sigma^2} \right) r_{k,k+1}.$$

These are clearly analytically intractable, presenting the drawback that the equations require numerical solution and therefore incur additional computational expense. (In fact, in this case, the additional expense is fairly minimal (see Section 7.1), but this may not generally be the case.)

An alternative is to consider a second-order term and to set the first- and second-order derivatives of A to 0 at the non-identifiability centring point $u = 0$, ignoring the zeroth-order term. Simultaneously solving these two equations, we obtain the value of μ given in equation (7) and

$$\sigma^2 = \frac{\sigma_\varepsilon^2}{\sum_{t=k_{\max}+1}^{T} x_{t-k-1}^2 + \sigma_\varepsilon^2/\sigma_a^2}.$$

These values for μ and σ^2 both have plausible statistical interpretations. For the second-order method, μ is a function of the estimated squared correlation coefficient, of order k, between the residuals from the fitted AR(k) model. Recall that the squared correlation coefficient determines the maximum likelihood estimate for a_{k+1} and so the proposal is approximately centred at the maximum likelihood estimate. However, the variance σ^2 is the ratio between the model variance and the estimated variance. If the data are not particularly informative then the proposal variance increases.

We note also that the second-order proposal is the conditional posterior distribution of the new parameter a_{k+1} under the larger model M_{k+1} conditioning on the remaining parameters being unchanged. This corroborates the empirical observations of Troughton and Godsill (2001), who suggested that using the conditional distribution to propose the value of new parameters is particularly efficient and they went on to show that the proposal and acceptance ratio can be simplified in this case. We also note that this result generalizes to jumps between models differing by more than one dimension. For example, if we wish to move from model M_k to model $M_{k'}$ where $k, k' \in \{1, \ldots, k_{\max}\}$ and $k' > k$, using a multivariate normal proposal for the new variables (as suggested by Troughton and Godsill (2001)), then the second-order method provides the posterior conditional distribution of these new parameters conditioning on the rest, which remain unchanged. Of course, these methods may be extended further (Ehlers and Brooks, 2002) to the case in which the moves to smaller dimensional models are not deterministic, by simply differentiating the acceptance ratio with respect to all the u-terms that appear in either the numerator or denominator. See Brooks *et al.* (2003) for discussion of this generalization beyond the Bayesian model determination context.

Finally, if we wish to consider the conditional maximization method, we take the μ given in equation (7) and by centring at this point and using the zeroth-order method we obtain

$$\sigma^2 = \sigma_a^2 \left\{ \frac{r(k, k+1)}{r(k+1, k)} \right\}^2 \exp\left\{ \frac{-\mu_u \sum_{t=k_{\max}}^{T} \left(x_t - \sum_{\tau=1}^{k} a_\tau x_{t-\tau} \right) x_{t-k-1}}{\sigma_\varepsilon^2} \right\}.$$

Thus, we obtain a variance term that is similar to the zeroth-order solution.

4.4. In support of first- and higher order methods

For Euclidean state spaces, Langevin algorithms can be shown to have large computational advantages over corresponding zeroth-order methods in high dimensional problems; see Roberts and Rosenthal (1998). However, it is difficult to prove rigorous results to support the use of the first- and second-order methods in the general framework of this paper. Nevertheless, it is possible to make some progress, at least in simple stylized examples. Even for these toy examples, the results do not appear to be totally intuitive, and they give support to the notion of attempting to construct algorithms which have little variation in acceptance probabilities (i.e. having first- and higher order properties but not necessarily zeroth-order properties). This idea is verified empirically in the autoregressive case by the results provided in Section 7.1.

The analysis will use a notion called *capacitance* (Lawler and Sokal, 1988), which is well known to be closely related to the rate of convergence of a Markov chain by *Cheeger's inequality* as we shall see. We define the capacitance of a reversible Markov chain by

$$\kappa = \inf_A \left\{ \int_A \frac{\pi(\mathrm{d}x)\, P(x, A^c)}{\pi(A)} \right\} = \inf_A \{\kappa(A)\}, \tag{17}$$

say, where the infimum is taken over all measurable sets A such that $\pi(A) \leqslant \frac{1}{2}$. Here $P(x, A^c) = \mathbb{P}(X_1 \in A^c | X_0 = x)$ for a Markov chain X and $\kappa(A)$ just describes the probability of moving from A to A^c, for a chain started at stationarity within A. Cheeger's inequality tells us that, if r is the supremum of the spectrum of the Markov chain transition operator (i.e. usually its rate of convergence), then

$$1 - 2\kappa \leqslant r \leqslant 1 - \kappa^2/2.$$

See Brooks and Roberts (1999), for example in the context of MCMC methods.

This result tells us that a surrogate for convergence of a Markov chain is its capacitance, although we cannot precisely identify a chain's rate of convergence from its capacitance. Therefore, it is natural to attempt to construct algorithms with the largest possible capacitance (decreasing both the lower and the upper bounds on the convergence rate) and in simple reversible jump settings we can identify these algorithms and characterize them in terms of first-order properties of the Markov chain.

The following result returns to the example of Section 3.2 in the context of which we can discuss reversible jump and our methodology in a non-trivial situation, where transitions are being constructed between spaces of dimension 0 and 1.

Lemma 1. Suppose that $\Theta = \Theta_1 \cup \Theta_2$ with $\Theta_1 = \{e\}$ (where e is some arbitrary singleton not contained in $[0, 1]$) and $\Theta_2 = [0, 1]$. Let $\pi(e) = p$ and $\pi(u) = (1 - p)\, f(u)$ for some prob-

ability density function f on [0, 1], i.e. $\pi(x) = p\,\delta_e(x) + (1-p)\,f(x)$ where $\delta_e(\cdot)$ denotes a point of unit mass at e. Consider the algorithm which always attempts to jump from the current model to the other (i.e. without any within-model moves). Let the proposal density be q if we are currently in space M_1 attempting a move to the one-dimensional space M_2, and all moves from M_2 just attempt to jump to e. Then, among all possible choices of q, the capacitance of the algorithm is maximized by the choice $q = f$.

The proof is given in Appendix B.

Remark 1. In the language of this paper, this result can be restated as saying that the algorithm which maximizes capacitance is kth order for all $k \geqslant 1$, but not for $k = 0$. This is because the maximizing algorithm (with $q = f$) leads to an acceptance ratio of $A_{1,2} = (1-p)/p$. Clearly, the derivatives of this are all 0, but the acceptance rate itself will only be 1 if $p = \frac{1}{2}$. Thus, the optimal algorithm is kth order for $k \geqslant 1$, but not zeroth order. A first- and/or second-order method may be a good approximation to this optimal algorithm in specific cases.

An alternative to the kth-order approaches described so far can be developed by adopting a saturated space approach which allows the chain to retain information when going from a larger to a smaller model which can be used when returning to that model later. This approach is discussed in the next section.

5. The saturated space approach

The saturated space approach involves augmenting the state space of the Markov chain so that the dimension of the chain remains constant throughout the simulation. At any given time, some of the states of the chain will correspond to model parameters and the rest can be used to retain information about where the chain has been in the past. In particular, if we move from a larger to a smaller dimensional model, information can be retained so that, when we return to the larger model later in the simulation, we can use this information to ensure that we propose jumping to a sensible place.

We shall introduce the saturated approach in the context of arbitrary reversible jump dynamics, so that moves to lower dimensional spaces are no longer confined to be deterministic, as is assumed in Sections 2 and 3.

5.1. Augmenting the state space

Suppose that $\sup_i(n_i) = n_{\max} < \infty$ and let $(M_i, \boldsymbol{\theta}_i)$ denote a random variable distributed according to π. Define a collection of dual random variables conditional on the value of $(M_i, \boldsymbol{\theta}_i)$ in the following way. Given $(M_i, \boldsymbol{\theta}_i)$, let $u_{i,r}, n_i + 1 \leqslant r \leqslant n_{\max}$ be a collection of univariate random variables with joint density $q_{n_{\max}-n_i}$ with respect to $(n_{\max} - n_i)$-dimensional Lebesgue measure and which is independent of the current value of $\boldsymbol{\theta}_i$ except its dimensionality. We shall write $\mathbf{u}_i = (u_{i,n_i+1}, \ldots, u_{i,n_{\max}})$. Also suppose that $w_k, 1 \leqslant k \leqslant n_{\max}$, are an independent and identically distributed collection of random seeds each drawn from density $r(\cdot)$. The us play the role of 'dimension saturation', whereas the ws provide a source of additional randomness in the between-model move.

Given M_i, $(\boldsymbol{\theta}_i, \mathbf{u}_i)$ describes an $n_{\max}$-dimensional random vector with joint density with respect to $n_{\max}$-dimensional Lebesgue measure given by

$$\pi_{\text{aux}} = \pi(M_i, \boldsymbol{\theta}_i)\, q_{n_{\max}-n_i}(\mathbf{u}_i) \prod_{k=1}^{s} r(w_k) \tag{18}$$

for some integer $s \leqslant n_{\max}$. This approach is similar to that described by Besag (2000), but we use this set-up to construct more flexible families of algorithms through the use of dependence structures between AVs.

The saturated space approach is quite different from the product form construction that was introduced by Carlin and Chib (1995), though it has some similarities to the generalization of that approach described by Godsill (2001). Our approach differs from that of Godsill (2001) in that the dimension compensating components stored are not specific to a particular statistical parameter and can be used to generate moves to different (perhaps non-nested) models. In the nested case, our approach is very similar to that of Godsill (2001). Here we add sufficient AVs to augment the dimensionality of the space to equal that under the 'largest' model under consideration. Note the distinction from the product space construction which requires the algorithm simultaneously to store a parameter vector for each model under consideration. The saturated space approach also removes the need for the pseudopriors that are necessary for the product space implementation, as well as having the obvious computational advantage in that far fewer AVs are required.

Under this saturated space arrangement, the Markov chain simulation proceeds in three stages at each update. First we update the states corresponding to the model parameters under the current model, i.e. $\boldsymbol{\theta}_i$. This can done in the usual way by using some form of Metropolis–Hastings update for example. Next we update the elements of $\mathbf{u}_i$ and $\mathbf{w}$ by any procedure which preserves the stationary distribution of $\mathbf{u}_i$ and $\mathbf{w}$ conditionally on $\boldsymbol{\theta}_i$. Finally, we update the model by using a reversible jump step, which is now of fixed dimension.

The most straightforward way of updating the random seeds $\mathbf{u}_i$ and $\mathbf{w}$ is to replace them with independent draws from their known distributions but, as we shall see in Section 6, other interesting alternatives are available. However, here we shall restrict ourselves to the case where the random seeds are updated independently of their previous values.

Here there are different bijective maps between model spaces depending on the random seed $\mathbf{w}$ that is chosen. These we denote by $h^{\mathbf{w}}_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i)$ with the acceptance probability of moves still described by equation (6) (with a single superscript added to h in the Jacobian term).

5.1.1. *Example: autoregressive example revisited*

Here we might choose to take $h^{\mathbf{w}}_{i,i+1}(\boldsymbol{\theta}_i, \mathbf{u}_i) = (\boldsymbol{\theta}_i + (w_1, \ldots, w_i), Ru_{n_{\max}-i})$ for some appropriate scaling constant R. The choice of R can be decided by using an appropriate zeroth-order method satisfying equation (8). Note that, although centring functions can also be chosen to be dependent on $\mathbf{w}$, we shall assume that this is not the case. So the simplest possible zeroth-order algorithm just fixes $\mathbf{w}$ and scales $\mathbf{v}_i$ as in Section 3. Of course it is possible to come up with more complex choices for satisfying equation (8) which scales $\mathbf{w}$ as well as $\mathbf{v}$.

6. Serially correlated random seeds

The retention of the random seeds that are used to update the Markov chain allows more flexible move types to be easily constructed here. We shall introduce a collection of AV methods within the saturated space framework. With these methods, mixing around model spaces can be assisted by 'momentum' induced through the AVs themselves. Within Euclidean and simple finite space contexts, there is considerable empirical and theoretical evidence for the effectiveness of similarly motivated AV methods (see for example Duane *et al.* (1987) and Diaconis *et al.* (2000)).

We now introduce two examples of how the increased flexibility of the saturated space approach can be used to construct new Markov chain dynamics which have the potential to

provide improved between-model communication. We describe two distinct applications of the saturated state space approach. The first introduces a memory property to the chain by directly inducing temporal dependence between the **u**-vectors through the use of an autoregressive updating scheme which retains complete independence between elements of the **u**-vector. The second creates a form of momentum by introducing dependence between the elements of the **u**-vector and giving the chain a propensity to make certain types of move for a period of time. We begin by describing the first, which we call the *independent* AV method. Since none of our examples need the extra generality, for simplicity of notation we shall avoid the use of **w** as introduced in Section 5 by setting $s = 0$.

6.1. The independent auxiliary variable method

The vanilla reversible jump algorithm proceeds by generating the elements of the **u** independently both of one another and of the values generated for previous iterations. In this subsection, we begin by introducing temporal dependence between the **u**-vectors, through the use of an autoregressive updating procedure. In practice, this provides the algorithm with a form of short-term memory. Essentially, when we move from a larger to a smaller model, information regarding the final position in the larger model is stored in the **u**-vector. If the reverse move is proposed, then this information can be used to ensure that a sensible jump is proposed. However, alternative moves (perhaps to a different mode, for example) may also be desirable, and so this memory is designed to be short lived, essentially reverting to the vanilla reversible jump scheme over time if the reverse move fails to be accepted. Obviously, this autoregressive updating scheme induces slower convergence for the AVs and there is, therefore, a trade-off between the length of the memory property of the algorithm and the ability to propose moves to entirely new places in the larger model.

Here, we suppose that, given $(M_i, \boldsymbol{\theta}_i)$, the $u_r,\ n_i + 1 \leqslant r \leqslant n_{\max}$ are a collection of independent univariate random variables each with density q with respect to Lebesgue measure. Thus, the joint density in equation (18) becomes

$$\pi_{\text{aux}}(M_i, \boldsymbol{\theta}_i, \mathbf{u}_i) = \pi(M_i, \boldsymbol{\theta}_i) \prod_{k=n_i+1}^{n_{\max}} q(u_{i,k}).$$

Suppose that we have a collection of injective maps $\{h_{i,j} : (i, j) \in E\}$ where $E = \{(i, j) : r_{ij}(\boldsymbol{\theta}_i) > 0 \text{ for some } \boldsymbol{\theta}_i \in \Theta_i\}$ denotes the set of pairs (i, j) for which jumps between models M_i and M_j are allowed. For a particular pair (i, j) such that $n_i < n_j$, the jump function $h_{i,j} : \Theta_i \times \mathbb{R}^{n_{\max}-n_i} \to \Theta_j \times \mathbb{R}^{n_{\max}-n_j}$ is a bijective map that fixes $u_r, n_j + 1 \leqslant r \leqslant n_{\max}$, i.e. if $h_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i) = (\boldsymbol{\theta}_j, \mathbf{u}_j)$ then $u_{i,r} = u_{j,r} \forall n_j + 1 \leqslant r \leqslant n_{\max}$. Finally, we set $h_{j,i} = h_{i,j}^{-1}$.

Given that the current state of the Markov chain is $(M_i, \boldsymbol{\theta}_i, \mathbf{u}_i)$, then the algorithm chooses an element at random from $\{l : (i, l) \in E\}$, j say, and proposes the move to $h_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i)$ according to the probabilities $r_{i\cdot}(\boldsymbol{\theta}_i)$. The move is then accepted with probability $\alpha\{(M_i, \boldsymbol{\theta}_i, \mathbf{u}_i), (M_j, \boldsymbol{\theta}_j, \mathbf{u}_j)\} = \min\{1, A_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i; \boldsymbol{\theta}_j, \mathbf{u}_j)\}$, where

$$A_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i; \boldsymbol{\theta}_j, \mathbf{u}_j) = \frac{\pi(M_j, \boldsymbol{\theta}_j)\, r_{ji}(\boldsymbol{\theta}_j)}{\pi(M_i, \boldsymbol{\theta}_i)\, r_{ij}(\boldsymbol{\theta}_i) \prod_{r=n_i+1}^{n_j} q(u_{i,r})} |J_{i,j}^{h}(\boldsymbol{\theta}_i, u_{1,n_i+1}, \ldots, u_{i,n_{\max}})|. \tag{19}$$

Here $\boldsymbol{\theta}_j = h_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i)$. Note the strong similarity with the acceptance ratio given in equation (6). Obviously, for $n_i > n_j$ the acceptance probability is given by

$$\alpha\{(M_i, \boldsymbol{\theta}_i, \mathbf{u}_i), (M_j, \boldsymbol{\theta}_j, \mathbf{u}_j)\} = \min\{1, A_{j,i}^{-1}(\boldsymbol{\theta}_j, \mathbf{u}_j; \boldsymbol{\theta}_i, \mathbf{u}_i)\}.$$

Furthermore, jumps between models of the same dimensionality leave the AVs unchanged and so we just resort to standard Metropolis–Hastings transitions.

Combined with a positive recurrence property of all algorithms updating within each Θ_i, the overall algorithm is suitably positive recurrent. Here the $\mathbf{u}$ are playing the role of the random draws from the proposal distribution used to do reversible jump MCMC sampling. However, including them explicitly in the target density provides us with additional flexibility in constructing algorithms depending on how we update the $\mathbf{u}$. However, we can move beyond the vanilla reversible jump algorithm by adopting a Markov updating scheme for the elements of $\mathbf{u}$. This can be done in any manner of ways. We illustrate one approach in the context of the autoregressive example.

6.1.1. Example: autoregressive example revisited

We may update the elements of $\mathbf{u}$ in any manner which ensures that their stationary distribution is that specified for q. An alternative to the vanilla algorithm above is to use any Markov scheme which produces the correct stationary distribution. In many cases, Metropolis–Hastings moves may be used to update $\mathbf{u}$, but in other contexts more direct methods may be applied. If, for example, we choose the standard Gaussian distribution to be our stationary distribution q, then we might use an autoregressive process to update the elements of $\mathbf{u}$ as follows.

Suppose that we are currently in model M_k (where $k \leqslant k_{\max}$) and that we need to update each of the elements $u_{k,k+1}, \ldots, u_{k,k_{\max}}$. We may consider each in turn (since we are assuming that they are independently—and identically—distributed). Let us consider the update for $u_{k,r}$. If we take a new value for this variable ($u'_{k,r}$, say) such that

$$u'_{k,r} = \lambda u_{k,r} + N(0, 1 - \lambda^2),$$

for some $\lambda \in [-1, 1]$, then the stationary distribution of this process is the standard normal distribution which we would adopt as our density q. Of course, $u'_{k,r}$ need not exist on the whole real line, in which case an alternative process and stationary distribution would be required.

In the context of the autoregressive example and considering a jump from $(\boldsymbol{\theta}_k, \mathbf{u}_k)$ to $(\boldsymbol{\theta}_{k+1}, \mathbf{u}_{k+1})$, we might use a combination of the zeroth-order method and the uncorrelated AV method above, by setting $\theta_{k+1,k+1} = \sigma u_{k,k+1}$. Recall, that the zeroth-order method suggests scaling the standard normal distribution by σ, given in equation (14). Thus, the q-term in equation (12) is simply a normal density with zero mean and variance σ^2. Note also that when doing the reverse move we would set $u_{k,k+1} = \theta_{k+1,k+1}/\sigma$. Of course, this idea can easily be extended to the second-order method by simply setting $\theta_{k+1,k+1} = \mu + \sigma u_{k,k+1}$, for example.

The desirable properties of the uncorrelated AV described above can be plainly seen in the context of this example. Suppose that we go from model $k + 1$ to model k; then the old value of a_{k+1} is stored as $u_{k,k+1}$. As the simulation continues, this value will continue to be updated. However, if λ is close to 1, then movement will be very slow and, if the reverse jump is proposed (from k to $k + 1$) relatively quickly, then the proposal will be to move to somewhere close (in terms of the value of a_{k+1}) to where it was when it last left that model.

Thus, as described above, the uncorrelated AV algorithm provides the chain with a form of memory, making it easier to move between models. Of course, the length of this memory depends on the value of λ. The larger the value, the longer the memory. Obviously, if we implement the kth-order AV method, but take $\lambda = 0$, we obtain just the kth-order method. We examine the performance of these algorithms, as the value of λ varies, in Section 7.1.

6.2. The correlated auxiliary variable algorithm

In this section, we extend the uncorrelated AV method to introduce dependence between elements of the **u**-vector as well as the temporal dependence induced by the uncorrelated approach described above. Here, rather than the explicit use of the autoregressive updating scheme to induce temporal dependence, we use the single-parameter Gibbs sampler to update highly correlated AVs. Thus, we make use of the usually undesirable property that is inherent in the Gibbs sampler that individual updates of highly correlated parameters create a slowly mixing chain. See Roberts and Sahu (1997), for example. In addition, the introduction of a degree of correlation between the AVs can be used to encourage certain types of move at certain times. This introduces a kind of momentum since there will be periods in which the AVs are 'lined up' to promote (for instance) either model complexity or parsimony. This may be particularly useful in the presence of multimodality, as observed in the mixtures problem described in Section 9 for example.

The motivation behind the correlated AV approach is that often there are models of comparable complexity, but for which traversing between the two models is very difficult since intermediate states are very weakly supported by the data. However, traversing between these states might be made considerably easier by following a path through a collection of much simpler models. The correlated AV allows the Markov chain occasionally to make an excursion to an extremely simple model from which it might return to the 'other model' once complexity is restored. Thus, this method can be thought of as a kind of tempering (Marinari and Parisi, 1992), where the role of temperature is played by the propensity of the AVs to promote parsimony, and where the temperature change is assisted by the momentum introduced. The use of auxiliary momentum variables has been successfully used in other MCMC contexts, particularly in the physics literatures; see for example Duane *et al.* (1987) and Neal (1996). Diaconis *et al.* (2000) have given compelling theoretical arguments in toy examples for the usefulness of these techniques.

Though the method is more generally applicable, we restrict our attention to a Gaussian formulation for illustration. The basic idea is to assume that the $\mathbf{u}_{i,r}$ random variables are exchangeable with distribution $N(\mathbf{0}, \Sigma_i)$ where Σ_i denotes the i-dimensional covariance matrix in which all variables have unit variance and the covariance between any two is ρ. As before, the algorithm alternates between updating $\mathbf{u}_i$ according to any Markov chain which preserves $q_{n_{\max}-n_i}$ and proposing model jumps in the usual way. In this case equation (19) is modified slightly to give

$$A_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i; \boldsymbol{\theta}_j, \mathbf{u}_j) = \frac{\pi(M_j, \boldsymbol{\theta}_j) r_{ji}(\boldsymbol{\theta}_j) q_{n_{\max}-n_j}(u_{n_j+1}, \ldots, u_{n_{\max}})}{\pi(M_i, \boldsymbol{\theta}_i) r_{ij}(\boldsymbol{\theta}_i) q_{n_{\max}-n_i}(u_{n_i+1}, \ldots, u_{n_{\max}})} |J^h_{i,j}(\boldsymbol{\theta}_i, u_{i,n_i+1}, \ldots, u_{i,n_j})|.$$

The steps used to update $\mathbf{u}_i$ can be carried out in a variety of ways. One natural scheme is to update the variables singly by Gibbs sampling. In this case,

$$u_{i,r}|\mathbf{u}_{i,(r)} \sim N\left[\frac{\rho \sum_{s \neq r} u_{i,s}}{1 + (d-2)\rho}, \frac{(1-\rho)\{1 + (d-1)\rho\}}{1 + (d-2)\rho}\right],$$

where $\mathbf{u}_{i,(r)}$ denotes the vector $\mathbf{u}_i$ with the rth element removed and d is the dimension of the $\mathbf{u}_i$-vector, i.e. $d = n_{\max} - n_i$. As with the uncorrelated method, the correlated auxiliary and kth-order methods may be combined and we shall demonstrate this in Section 7.1.

Of course, it may not always be possible to bound the dimensionality of the 'largest' model

under consideration. In this case we need to extend the method described above to allow for the inclusion for a large or possibly infinite number of AVs.

6.3. The infinite correlated auxiliary variable method

Since $n_{\max}$ in Section 6.2 can be replaced by any larger integer, a natural possibility is to consider the case where we have an infinite collection of AVs. This allows us to consider the case where the dimensionality of the models under consideration is unbounded. By de Finetti's theorem, in the exchangeable case, this can be reformulated hierarchically as follows.

Suppose that Y is an $N(0, \rho)$ variable and, conditional on Y, we set each $u_{i,r}$ to be independently $N(Y, 1-\rho)$. This formulation is particularly attractive for the correlated AV method, since we can assume that we have this infinite collection $\mathbf{u}$, though we store only as many as we need. If we need a new one, we just generate it from its distribution conditional on Y. In this case the acceptance ratio for a move from smaller dimension n_i to larger dimension n_j is described by

$$A_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i; \boldsymbol{\theta}_j, \mathbf{u}_j) = \frac{\pi(M_j, \boldsymbol{\theta}_j)\, r_{ji}(\boldsymbol{\theta}_j)\, \tilde{q}_{n_{\max}-n_j}(u_{n_j+1}, \ldots, u_{n_{\max}})}{\pi(M_i, \boldsymbol{\theta}_i)\, r_{ij}(\boldsymbol{\theta}_i)\, \tilde{q}_{n_{\max}-n_i}(u_{n_i+1}, \ldots, u_{n_{\max}})} |J^h_{i,j}(\boldsymbol{\theta}_i, u_{i,n_i+1}, \ldots, u_{i,n_j})|, \tag{20}$$

where $\tilde{q}_d$ denotes the density of d independent random variables with distribution $N(Y, 1-\rho)$.

The algorithm therefore proceeds as follows. Suppose that we are currently at model M_i, and in state $(\boldsymbol{\theta}_i, \mathbf{u}_i)$.

(a) Update Y according to any Markov chain dynamic preserving its distribution.
(b) Choose a model to try to move to according to $r_{i\cdot}(\boldsymbol{\theta}_i)$, j say.
(c) Suppose that $n_j > n_i$; then
  (i) generate $u_{n_i+1}, \ldots, u_{n_j}$ according to $N(Y, 1-\rho)$,
  (ii) accept the move to $h_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i)$ with probability $\min\{1, A_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i; \boldsymbol{\theta}_j, \mathbf{u}_j)\}$ given in equation (20) or
  (iii) otherwise remain at $(M_i, \boldsymbol{\theta}_i, \mathbf{u}_i)$.
(d) If $n_j < n_i$, compute $h^{-1}_{i,j}(\boldsymbol{\theta}_i, \mathbf{u}_i)$, which gives us $(\boldsymbol{\theta}_j, \mathbf{u}_j)$. Accept this move with probability $\min\{1, A_{j,i}(\boldsymbol{\theta}_j, \mathbf{u}_j; \boldsymbol{\theta}_i, \mathbf{u}_j)\}^{-1}$. Otherwise stay at $(M_i, \boldsymbol{\theta}_i, \mathbf{u}_i)$.

6.4. Generalizing the auxiliary variable methods

The examples that are provided in this section are based on the assumption that our proposal q is of (multivariate) normal form. In many cases, this will not be true and so we require the introduction of a more general method for producing dependent sequences with arbitrary stationary densities. This problem reduces to the case where q denotes a standard $(n_{\max} - n_i)$-dimensional uniform density, since, given a sequence of standard uniform vectors, $\mathbf{u}$-variates from any arbitrary density may be obtained (via inversion, for example). Thus, without loss of generality, we shall focus on the standard uniform case here. Given the examples already provided in this section, it is clear that such a sequence may be obtained simply by taking the inverse normal cumulative density function of the $\mathbf{u}$-variates described above. However, a simpler method is to construct a scheme which induces a stationary uniform density directly. One such scheme is the so-called 'moody ring' scheme described below.

Suppose that we need to update $n_{\max}$ parameters of which n_i are associated with the current model and $n_{\max} - n_i$ are the AVs. Thus, $\mathbf{u}_i = (u_{n_i+1}, \ldots, u_{n_{\max}})$. It is easiest to begin with the correlated AV method for which we require an updating scheme which induces dependence between both successive iterations and across (future) components. We also require that the

process has a stationary marginal distribution which is the standard uniform distribution for all elements of the **u**-vector.

Consider a process $\{C^t\}$ for which we set $C^t = (C^{t-1} + w^t)_{\text{mod } 1}$ where $w^t \sim U[-\varepsilon, \varepsilon]$, $\varepsilon \leqslant 0.5$. This essentially describes a random walk on [0, 1] in which the end points are joined. You might think of C^t as living on a ring of circumference 1, where the value of C^t is determined by the distance along the ring (in a clockwise direction) from a fixed point on the circumference. The stationary distribution for C^t is therefore uniform on [0, 1]. We call C^t the mood parameter, since its value will typically favour one particular move (such as one which increases the dimension—see Section 9) over any other at any particular time. Given the mood parameter C^t, we can generate the elements of $\mathbf{u}_i^t$ (given by u_l^t, $l = 1, n_{\max} - n_i$) by setting $u_l^t = (C^t + z_l^t)_{\text{mod } 1}$ where $z_l^t \sim U[-\delta, \delta]$, $\delta \leqslant 0.5$, $l = 1, \ldots, n_{\max} - n_i$.

The fact that all the u_l are generated from the same distribution induces a dependence between them which increases as $\delta \to 0$. Thus, for small δ, whatever 'mood' the C^t-chain is in, the u_l^t-chains will all be in a similar mood. In addition, the mood parameter moves around the ring, inducing a dependence across iterations, the strength of which depends on the value of ε. If both δ and ε are small, then moods (values) will be consistent across the $\mathbf{u}_i$-vector and mood changes will be slow across time. This means that we induce prolonged periods in which, for example, dimension-changing moves are easy to perform, and thus allows greater opportunity for chains to move between models of appreciable probability mass separated by more than one reversible jump move in which intermediate models are not well supported. An example illustrating this behaviour is provided in Section 9.

For the uncorrelated case, we simply remove the mood parameter and update the u_l^t independently of one another, so that $u_l^t = (u_l^{t-1} + z_l^t)_{\text{mod } 1}$ so that each u_l^t follows its own random walk around the ring over time. If we set $\varepsilon = \delta = 0.5$ we obtain the vanilla reversible jump MCMC algorithm.

7. Comparing the methods

In this and the following sections, we illustrate our methodology with the aid of three examples which have been chosen to provide a representative sample from the range of problems to which the reversible jump MCMC technique has been regularly applied. However, before we do so, we begin by discussing a variety of methods which can be used to assess the performance of the sampler so that the various algorithms may be compared.

The performance assessment techniques can be split into two categories: the numerical and the graphical. Graphical techniques include raw trace plots, autocorrelation plots and cumulative plots. Since these are to be plotted over all iterations, we must first find statistics to plot which retain a constant interpretation across all models. In most cases, the model number may be plotted. For example in the autoregressive case the model number is simply the autoregressive order, in the mixtures case it would be the number of components and in the graphical models case it may be the number of edges or some lexicographic representation of the current graph. An alternative is to plot the deviance over time as that also retains a constant interpretation. Examples of cumulative plots include the cumulative number of models visited within a simulation, and the cumulative occupancy fractions (see Richardson and Green (1997), Brooks and Giudici (2000) and Brooks *et al.* (2002)).

Numerical assessment techniques include monitoring acceptance rates for model-changing moves, noting the total number of models visited, effective sample size (ESS) calculations and convergence rate estimates. Acceptance rates for reversible jump MCMC moves are typically somewhat lower than those for fixed dimension Metropolis–Hastings moves, for example.

Though a high acceptance rate does not necessarily guarantee good sampler performance (Gelman *et al.*, 1996) an increase in the acceptance rate while retaining the same posterior inference would usually be viewed as an improvement. ESS calculations can be obtained for statistics that retain a coherent interpretation throughout the simulation. These tell us how many independent observations are equivalent (in terms of learning about specific statistics of interest) to the set of dependent observations actually obtained. Thus, the larger the ESS is, the better the performance of the algorithm; see for example Hastings (1970). Perhaps most useful are comparisons of ESS per second (Sargent *et al.*, 2000) which also incorporate computational expense for a more practical comparison. The simplest comparison of this form can be made by recording the model order variable throughout the simulation and comparing the ESS (in terms of the mean of this variable) across the various simulations. Simulations which mix better in terms of movement between models will have smaller autocorrelation times and therefore larger ESSs. See Brooks and Giudici (2000), for example. Finally, convergence rate estimation (in terms of the marginal distribution over the model space) may be obtained by examining the marginal distribution of the model number (or any other scalar statistic with constant interpretation, such as the number of edges) and deriving an empirical transition matrix for this sub-Markov chain. The second largest eigenvalue of this matrix provides an indication of the convergence rate; see Brooks *et al.* (2002).

None of the methods described above provide a reliable comparison in themselves, but together they provide sufficient information to begin to make some general statements comparing two or more samplers. We hope that in computing a variety of performance statistics we can draw fairly broad conclusions about the relative merits of competing algorithms. Obviously there is considerable scope for future work in this area.

7.1. Example: autoregressive model choice

Throughout the preceding sections, we have illustrated our methods with reference to the analysis of autoregressive time series. We begin our discussion, comparing the various methods, by examining their performance in the context of this example.

We consider an analysis of the data described and modelled by Huerta and West (1999). This series consists of 540 monthly observations of the southern oscillation index during 1950–1995, measuring the 'difference of the departure from the long-term monthly mean sea-level pressures' at Tahiti in the South Pacific and Darwin in Northern Australia. For each method we take an $N(0, 1)$ prior for the autoregressive parameters, a $\Gamma^{-1}(10^{-3}, 10^{-3})$ prior for the error variance and a uniform prior on values of k from 1 to 10. These priors are chosen to be reasonably vague and to provide an acceptable compromise in terms of their influence on the model parameters and the models themselves. See Jennison (1997) and Berger (2000), for example.

For each method, we run three independent replications of 1 million iterations, thinning to every 10th value to reduce computational overheads associated with storage. In all simulations, the within-model parameter estimates were essentially identical and each simulation gave identical orderings of the models in terms of posterior model probabilities, though there was some variation in the actual posterior probability values obtained. Estimates of the Monte Carlo standard error of the posterior probability for the most likely model (*a posteriori*) are around 0.003 for all simulation algorithms. All methods attributed the highest posterior probability to the AR(3) model with significant posterior mass placed also on AR(2) and steadily decreasing probability assigned to higher order models, as we would expect (Ehlers and Brooks, 2002).

For illustration, we ran the vanilla algorithm with a pilot-tuned proposal distribution for model moves fixed to be a normal distribution with mean 0 and variance 0.01. These values

were obtained by using the standard practice of running a variety of simulations with different values and choosing the set which gives the highest acceptance rates. For comparison, we also ran the zeroth-order, first-order, second-order and conditional maximization methods together with a series of AV methods. We begin with the vanilla uncorrelated AV method (i.e. the correlated AV method with no kth-order methodology), taking a range of λ-values. We then fixed λ to be 0.5 and took a combination of the uncorrelated AV method with the zeroth-order, first-order, second-order and conditional maximization methods. Finally, we ran a similar range of simulations for the correlated AV method. The results of these simulations are presented in Table 1.

We can see from Table 1 that all methods visit at least models 2–6. Although there is some variability between the value of the highest posterior model probability, this is well within the range expected given the (fairly low) Monte Carlo standard errors which are, as we would expect, larger for those algorithms performing least well.

In terms of the acceptance rate, the first-order, second-order and conditional maximization methods appear to perform well, with a twofold improvement over the vanilla method. Similarly, the ESSs demonstrate a threefold improvement over the vanilla algorithm and the estimated convergence rate an approximately twofold increase in performance. This improvement appears to be at the expense of a modest 10% increase in computation.

Table 1. Summary statistics for the autoregressive example: acceptance rate $\bar{\alpha}$, range of models visited, posterior probability of the 'true' model, ESS (from a thinned sample of size 100000), computation time and estimated convergence rate (for the thinned sample) $\hat{r}$†

Method		$\bar{\alpha}$	*Models*	$\pi(M_3)$‡	*ESS*	*Time (s)*	$\hat{r}$
Vanilla		0.091	1–8	0.610	3878	412	0.776
Zeroth order		0.051	1–6	0.611	4887	440	0.848
First order		0.203	2–8	0.610	10668	492	0.539
Second order		0.206	2–9	0.609	10850	464	0.545
CM		0.205	1–8	0.612	10268	444	0.573
	$\lambda = 0.2$	0.093	1–8	0.612	3773	402	0.808
	$\lambda = 0.5$	0.092	1–9	0.606	3571	399	0.889
UAV	$\lambda = 0.7$	0.091	1–8	0.610	3515	403	0.909
	$\lambda = 0.9$	0.091	1–7	0.614	2948	402	0.950
	$\lambda = 0.95$	0.091	1–7	0.610	2197	401	0.969
UAV (0)	$\lambda = 0.5$	0.051	1–5	0.616	4671	441	0.844
UAV (1)	$\lambda = 0.5$	0.204	2–8	0.615	9661	488	0.557
UAV (2)	$\lambda = 0.5$	0.206	2–8	0.616	10542	464	0.562
UAV (CM)	$\lambda = 0.5$	0.205	1–10	0.613	10137	450	0.588
	$\lambda = 0.2$	0.088	1–8	0.612	3471	406	0.778
	$\lambda = 0.5$	0.075	1–7	0.606	3359	408	0.852
CAV	$\lambda = 0.7$	0.062	2–7	0.614	2857	404	0.906
	$\lambda = 0.9$	0.040	2–7	0.609	1510	410	0.928
	$\lambda = 0.95$	0.030	2–7	0.607	1020	402	0.938
CAV (0)	$\rho = 0.5$	0.050	2–7	0.609	4634	468	0.875
CAV (1)	$\rho = 0.5$	0.165	2–7	0.606	9709	500	0.584
CAV (2)	$\rho = 0.5$	0.183	2–8	0.613	10525	481	0.596
CAV (CM)	$\rho = 0.5$	0.178	1–8	0.607	9729	462	0.613

†The performance is averaged over three replications of the different methods for the thinned chains. UAV, uncorrelated AV; CAV, correlated AV; CM, conditional maximization.
‡$\pi(M_3)$ denotes the estimated posterior probability associated with model M_3. Monte Carlo standard errors for these were also calculated from multiple replications of each algorithm and were between 0.002 and 0.004 for all simulations.

Turning to the uncorrelated AV method, we can see that increasing λ for the vanilla method results in a decreasing convergence rate and ESS, as we would expect, and that the acceptance rate appears to change very little. It is worth noting here that this is a particularly simple example. There is no multimodality in the model space and so the addition of the AVs would not be expected to improve on the vanilla algorithm, which itself performs very well on this example.

When we fix the dependence of the uncorrelated AV method and consider combining it with the kth-order methods, we observe a pattern similar to those without AVs. The first-order, second-order and conditional maximization methods appear to perform well and clearly better than the pilot-tuned vanilla algorithm. Similarly, when we examine the correlated AV method, we observe a similar performance to the uncorrelated AV method, though the improvement of the higher order methods over the vanilla method is lessened.

These results demonstrate that higher order methods appear to work at least as well as (if not better than) the pilot-tuned vanilla algorithm in this simple example. They therefore represent a considerable improvement over the vanilla algorithm since a comparable performance is obtained without the need for an expensive pilot tuning process. There appears to be no detectable additional benefit to the introduction of AV methods in this case. In the next two sections, we consider two further (and more challenging) problems and show how the kth-order and AV methods can make dramatic improvements in performance over pilot-tuned vanilla methods.

8. Graphical Gaussian models

Let $\mathbf{X}$ be a k-dimensional vector of random variables. A conditional independence graph $g = (V, E)$ describes the association structure of $\mathbf{X}$ by means of a graph, specified by the vertex set V and the edge set E. A *graphical model* is a family of probability distributions P_g which is Markov over g (see, for instance, Lauritzen (1996)). A graphical Gaussian model is obtained when only continuous random variables are considered and assuming $P_g = N(\mu, \Sigma_g)$, with Σ_g positive definite and such that P_g is Markov over g.

Recently, Giudici and Green (1999) proposed a hierarchical class of prior distributions and a reversible jump MCMC method, to perform both model selection and inference on the quantities of interest. At each stage, moves are performed, by adding or deleting one edge from the current conditional independence graph of the model, g, and checking that the resulting new graph g' is decomposable. When a new edge, say (i, j), is proposed for insertion, the dimensionality of the parameter space increases by 1; this implies the presence of an extra free element in Σ_g, σ'_{ij}. A realization of the new parameter element is sampled by drawing a random variable v from an $N(0, \sigma_g^2)$ distribution and setting $\sigma'_{ij} = v$. This proposal does not take into account the previous (constrained) state of σ_{ij}.

One difficulty with this approach is the choice of the spread parameter σ_g^2 of the proposal distribution. In Giudici and Green (1999) the constant was fixed, on the basis of several pilot runs, to be equal to $0.5n/|V|$, where n is the sample size and $|V|$ the cardinality of the vertex set V. Our aim here is to construct efficient reversible jump rules for the varying-dimension move, according to the recipes specified in the previous sections.

First, we note that the dimension matching constraint is specified by a 1–1 function between $\Theta = \{\sigma_{lk} : (l, k) \in E_g\}$ and $\Psi = \{(\sigma_{lk} : (l, k) \in E_g) \cap (\sigma_{ij} : (i, j) \notin E_g \cap (i, j) \in E_{g'})\}$. Therefore, the zeroth-order proposal leads us to set the proposal parameter to be given by

$$\sigma_g^{-1} = \sqrt{(2\pi)}\, \frac{h(\Sigma_S)\, h(\Sigma'_{Sij}, u = 0)}{h(\Sigma_{Si})\, h(\Sigma_{Sj})},$$

using the weak non-identifiability centring point for which $\sigma_{ij} = 0$.

The graphical Gaussian model example is highly complex and a very large family of possible models lies within the support of the posterior distribution. This makes pilot tuning of algorithm parameters particularly problematic. However, the zeroth-order method provides an automatic way of adaptively tuning the proposals, greatly reducing the time that is required to obtain reliable results. The large cardinality and potential multimodality of the model space suggests that, for graphical models, an AV scheme, such as those illustrated in Section 6, may improve the convergence.

The total number of AVs here is equal to the maximum number of edges possible, i.e. the number of edges in the complete graph, which we denote by $n_{\max} = n(n-1)/2$. To implement the saturated space approach, we assume that we have a vector of AVs, $u_1, \ldots, u_{n_{\max}}$, that we assume is distributed as multivariate Gaussian, with zero mean and variance–covariance matrix equal to an intraclass correlation structure

$$\Phi = \tau\{\rho J + (1-\rho)I\}, \tag{21}$$

where J is the $p \times p$ matrix of 1s and I the identity matrix of order p. We take $\tau = 1$.

In the MCMC implementation we sample each u_i from a proposal distribution corresponding to the full conditionals derived from the previous stationary distribution of the us. It is easy to derive the fact that the full conditionals are Gaussian with mean equal to $(\rho\Sigma_{j\neq i}u_j)/\{1 + (n_{\max} - 2)\rho\}$ and variance equal to $(1-\rho)\{1 + (n_{\max} - 1)\rho\}/\{1 + (n_{\max} - 2)\rho\}$.

8.1. Results

We first compare the mixing performance of our proposed zeroth-order algorithm with the vanilla (pilot-tuned) reversible jump scheme, as developed in Giudici and Green (1999). The comparisons will be made by using both graphical and more formal model convergence diagnostics. We remark that, for computational storage purposes, all graphical output has been thinned, retaining only one in every 10 observations. First we briefly consider, for illustration, one of the simplest, and most analysed, graphical modelling data sets: Fret's data, described in Whittaker (1990), concerning head measurements on pairs of sons in a sample of 25 families. In this example, since $k = 4$, the number of possible graphs is equal to 64, including three which are not decomposable.

We run two simulations each of length $n = 100\,000$, starting from the same point, for the vanilla and the zeroth-order methods. In reporting the results from MCMC model selection, we represent a graph by means of a vector of binary variables, indicating whether each edge is present (1) or absent (0), and with edges in a graph being ordered lexicographically. Fret's data contain at most six edges. The two graphs with the highest posterior probability are, with the vanilla method, (110111), with probability 0.13497, and (111011), with probability 0.12804. In other words, the two best graphs differ by the position of the chord that breaks the four-cycle. The zeroth-order method gives the same two best graphs: (110111) (0.12704) and (111011) (0.11728). However, the posterior probabilities are slightly lower in the zeroth-order case, suggesting that the Markov chain has spent more time in the tails of the distribution. In fact, on closer inspection, the vanilla algorithm visits only 23 distinct models compared with the zeroth-order method which sees a total of 29. This provides further evidence of the superiority of the zeroth-order method on this example.

We now consider the analysis of the fowl bones data set (Whittaker, 1990) concerning measurements on chicken bones. As there are six vertices, the number of possible graphs is 32 768. The total number of decomposable graphs is about 80% of these and so the resulting reversible jump MCMC simulation runs on a graph space with about 26 300 candidate models. Thus,

the model space is considerably larger than that for Fret's data. We investigate the zeroth-order method together with a correlated AV method in comparison with the usual (pilot-tuned) vanilla method.

An interesting problem here is that for one of the 15 edges the weak non-identifiability centring turns out to be inappropriate, precisely because the presence of the edge in question within the model is very strongly supported by the data. Thus, improvements on all the methods are possible by problem-specific centring strategies. Using the zeroth-order method for instance, the fact that the centring point is very much in the tail of the proposed model space leads to an extremely large variance for the proposed move. This is inappropriate in this example since what is being proposed is the introduction of a (non-null) partial correlation. Therefore, rather than refining the centring point (which would be a very problem-specific fix), we have imposed a truncation of the proposal variance to preclude the proposal of a large majority of impossible values for the partial correlation.

The first part of Table 2 demonstrates the substantial improvement in performance of the zeroth-order method over the vanilla method with a simulation run length of 1 million thinned to every 10th. The correlated AV method was implemented with $\rho = 0.5$ and, as we can see from Table 2, performed at least as well as the zeroth-order method.

Fig. 2 provides a trace plot of the number of edges, which can be taken as a measure of model complexity. The difference in performance between the vanilla and our methods is illustrated by the more rapid transitions between edge counts.

Fig. 3 compares more closely the behaviour of the posterior probabilities of the models. The plots give the cumulative number of different models visited by the Markov chain for the three algorithms. They demonstrate that the vanilla method has clearly failed to converge since it has visited not much more than half the number of models visited by the zeroth-order method or the correlated AV method. The correlated AV method visits many more models than the zeroth-order method also.

Looking more closely at the model posterior distributions, we found that all the methods find the two highest probability models in the same order (and appear to visit broadly the same class of popular models when we examine the ordered list of popular models in more detail). However, the second part of Table 2 shows that the zeroth-order and the AV method lead to a much more stable estimation of the posterior distribution.

In particular, the posterior probability variances and the distance measures presented in Table 2 are calculated on the basis of a multiple run of 10 chains with 1 million iterations, each started at a random point. The distance measure is obtained by considering, for each chain, the five most likely models and calculating, for each pair of chains, the number of matches. The final score for each method is obtained by summing the number of

Table 2. Fowl bones data: summary statistics (acceptance rate $\bar{\alpha}$ and ESS), model probability estimates (including the Monte Carlo variance) for the two models with greatest posterior mass and number of model matches between chains for thinned simulation output using different proposal determination methods

Method	$\bar{\alpha}$	*ESS*	$\pi(M_{30504})$	$var\{\pi(M_{30504})\}$	$\pi(M_{29992})$	$var\{\pi(M_{29992})\}$	*Distance*
Vanilla	0.001	91	0.17	0.10	0.001491	0.000268	0.747
Zeroth	0.015	874	0.18	0.11	0.000061	0.000062	0.449
CAV	0.026	1403	0.16	0.09	0.000174	0.000145	0.604

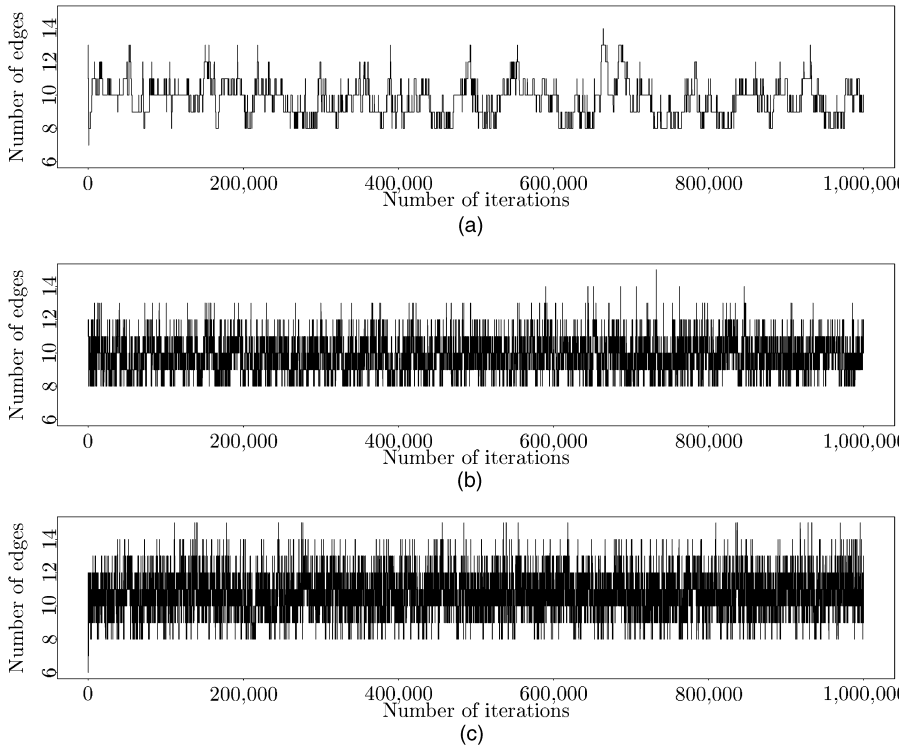

Fig. 2. Fowl bones data: diagnostic plots on the number of edges present for (a) the pilot-tuned vanilla algorithm, (b) the basic zeroth-order method and (c) the correlated AV method

matches over all pairs and normalizing by dividing it by the maximum possible number of matches.

To summarize our conclusions, the investigation of multiple long runs suggests reasonable stability in estimates from the zeroth-order and correlated AV method, and both of these appear to perform considerably better than the vanilla method.

9. Mixture models

As our final example, we re-examine the classification problem that was discussed by Richardson and Green (1997) and look at modelling a series of univariate data as a finite mixture of Gaussian distributions. This example presents peculiar difficulties. For instance, the model-changing moves generally involve moving between spaces differing by more than one dimension. Thus, we typically have more degrees of freedom in our proposal distribution for moving from one model to the next. Furthermore, the parameter spaces may be bounded both above and below if we impose (as is common) an ordering constraint on the component parameters. As a consequence, we are somewhat limited by analytic tractability in terms of the use of the kth-order methods. However, the AV methods remain very easy to use.

For an introduction to mixture modelling, we refer the reader to Richardson and Green (1997), who introduced a reversible jump MCMC scheme for a normal mixtures problem. Denoting the jth component by π_j with associated parameters μ_j and σ_j^2, the k-component normal

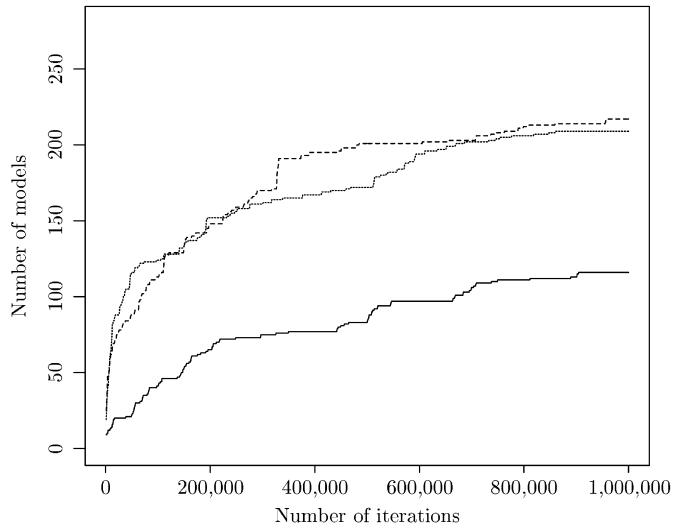

Fig. 3. Fowl bones data: cumulative number of models visited during simulation for the vanilla (———), zeroth-order (· · · · · · ·) and CAV (- - - - - - -) methods: the total numbers of models visited by these methods are 116, 209 and 245 respectively

mixture model is given by

$$f(x) = \sum_{j=1}^{k} w_j \, \pi_j(x).$$

Moves between models are performed via two reversible jump schemes known as split–combine and birth–death moves. We shall briefly consider the split–combine move here.

There are various ways in which the split–combine move may be implemented; however, Richardson and Green (1997) suggested the following. For the move which splits a single component (j) into two (j_1 and j_2), we adopt a moment matching strategy in which the weight that is assigned to that component is split between the two new components. Similarly, we assign means and variances to the new components which preserve the first- and second-order moments. This can be done by generating three random variables v_1, v_2 and v_3 from any density defined on [0, 1] and setting

$$w_{j_1} = w_j v_1,$$

$$w_{j_2} = w_j(1 - v_1),$$

$$\mu_{j_1} = \mu_j - v_2 \sigma_j \sqrt{\left(\frac{w_{j_2}}{w_{j_1}}\right)},$$

$$\mu_{j_2} = \mu_j + v_2 \sigma_j \sqrt{\left(\frac{w_{j_1}}{w_{j_2}}\right)},$$

$$\sigma_{j_1}^2 = v_3(1 - v_2^2)\sigma_j^2 \frac{w_j}{w_{j_1}},$$

$$\sigma_{j_2}^2 = (1 - v_3)(1 - v_2^2)\sigma_j^2 \frac{w_j}{w_{j_2}}.$$

Richardson and Green (1997) suggested generating the v_i from independent beta distributions and took

$$v_1 \sim \text{Be}(2, 2),$$

$$v_2 \sim \text{Be}(2, 2),$$

$$v_3 \sim \text{Be}(1, 1).$$

Let us begin by identifying the centring point for the move. Clearly, the weak non-identifiability centring point is obtained when the means and variances of the two new components are identical (and therefore the same as for the original component). This corresponds to $v_2 = 0$ and $v_1 = v_3$. Thus, the centring point is given by $(v_1, v_2, v_3) = (0.5, 0, 0.5)$, say. The interesting thing to note here is that the proposals taken by Richardson and Green (1997) for v_1 and v_3 have modes at the corresponding centre points, but this is not so for v_2. If we transform the proposal for v_2 so that $v_2 = |2v' - 1|$, with $v' \sim \text{Be}(2, 2)$, so that the mode is now at zero, then the acceptance rate for the split–combine moves increases from 8% (for the enzyme data set) to 10%, so we immediately appear to observe a small improvement just by thinking more carefully about the proposals. Note that the acceptance ratio A as defined in Richardson and Green (1997) is undefined if $v_2 = 0$, since v_2 appears explicitly in the denominator of the acceptance ratio. Although this does not affect the implementation of Richardson and Green's algorithm in its basic form, our kth-order methods may not be directly applied at the weak non-identifiability centring point. However, trivial manipulations allow us to rewrite the acceptance ratio in a form in which the v_2-term in the denominator cancels with a similar term in the numerator and so the kth-order methods may be applied. The obvious kth-order approach is to try to generate $v_i \sim \beta(\alpha_i, \beta_i)$ and to use perhaps the zeroth- to fifth-order equations to determine sensible values for these six parameters. A simpler alternative is to set $\alpha_i = \beta_i = R$ and to use just a single constraint to obtain R.

As a simple illustration suppose that we generate $u_i \sim U(0, 1)$ for $i = 1, 3$ and set $v(\mathbf{u}) = [\frac{1}{2} + (1 - 2u_1)R, 2Ru_2, \frac{1}{2} + (1 - 2u_3)R]$. The centring point is obviously $\mathbf{b} = (0.5, 0, 0.5)$, the proposal density $q(\mathbf{u}|R) = 1$ and the Jacobian term $|J^h| = (2R)^3$. Now, if we let g denote all the terms in the acceptance ratio in equation (6) except the proposal term, i.e.

$$g(\boldsymbol{\theta}_i, \mathbf{u}) = \frac{\pi(M_j, \boldsymbol{\theta}_j)\, r_{ji}(\boldsymbol{\theta}_j)}{\pi(M_i, \boldsymbol{\theta}_i)\, r_{ij}(\boldsymbol{\theta}_i)} |J^h(\boldsymbol{\theta}_i, \mathbf{u})|$$

where $\boldsymbol{\theta}_j = h_{ij}(\boldsymbol{\theta}_i, \mathbf{u})$, then the zeroth-order method gives the solution $R = g(\boldsymbol{\theta}_i, \mathbf{b})^{1/3}/8$. Compare this with equation (11), for example.

Using the zeroth-order method to determine the value of R for the enzyme data set of Richardson and Green (1997) and adopting their priors, the acceptance rate decreases the split–combine acceptance rate to 2.5% from the 8% acceptance rate that they observed. This does not improve over the results of Richardson and Green (1997), but this performance has been achieved without the need for pilot tuning of any kind.

The AV methods perform considerably better for this problem. However, since many methods perform adequately for the enzyme data problem, we shall artificially constrain the number of models, $k \leqslant 5$ (thus constraining to just over 80% of the posterior mass from the unconstrained problem), to see this advantage more clearly. The constraint has a serious effect on the mixing

of the vanilla algorithm since movement between parsimonious models that is well supported by the data cannot now be realized by transition through more complex models.

To examine the performance of the various methods, we ran each simulation for 200 000 iterations, discarding the initial 100 000 as part of the burn-in, and compared the vanilla algorithm, proposed by Richardson and Green (1997) on the basis of initial pilot tuning, with the uncorrelated AV and correlated AV algorithms using the moody ring method described in Section 6.4. Each algorithm was run with the same randomly chosen starting-points and pseudorandom seeds. We take $\varepsilon = 0.1$ and $\delta = 0.05$ and, though the output exhibits some sensitivity to the choice of these values, almost any values appear to provide an improvement over the vanilla method. Trace plots of the number of components are provided in Fig. 4 and these are typical of those observed for independent replications of the chains with different starting-points and seeds.

From Fig. 4 it is clear that the autocorrelated methods performed significantly better than the vanilla algorithm, which performs poorly. The AV methods are largely unaffected by the restriction on the model space (although they also suffer from the constraint to some extent). For example, the two-component model is visited only rarely by the vanilla algorithm (especially in the second half of the simulation) but much more regularly by the others. The corresponding acceptance rates for between-model moves are 4.3%, 17.7% and 5.3% for the vanilla, uncorrelated AV and correlated AV methods respectively. The slight 'blockiness' that is apparent in

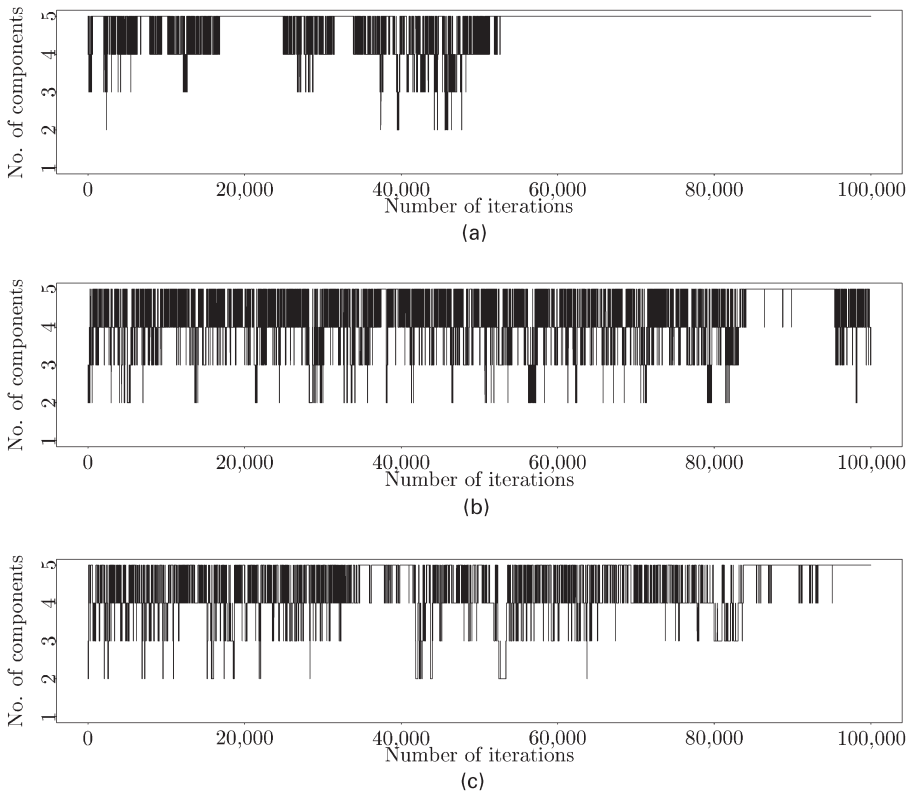

Fig. 4. Trace plots of the number of components for (a) the vanilla, (b) the uncorrelated and (c) the correlated AV methods for the mixtures example using the enzyme data set, considering only models with five components or fewer

the trace plot for the correlated AV method is due to the introduction of the central point. At certain positions on the ring, moves which increase the dimension are preferred, whereas, at other positions, moves which decrease the dimension are preferred. This leads to the slightly more 'blocky' trace plot in Fig. 4 and is an extremely useful property to have in the presence of extreme multimodality. The short-term persistence of particular move types in the correlated AV method allows the chain to explore further into the tails and provides greater potential for jumping between modes.

For this example, sensible monitoring procedures ought to pick up mixing problems in the vanilla algorithm, and it is clear how to improve mixing by increasing the permissible values of k. However, in general, it is difficult to determine how the range of models to be considered will affect the mixing properties of the corresponding algorithm. Indeed, in some cases, the range of models may be constrained to a small set by factors relating to the problem at hand. Thus, it is extremely difficult to predict whether or not the range of models allowed is sufficient to enable the vanilla algorithm to mix. Worse still, though for this problem the detection of inadequate mixing is easy, in general this will be far from obvious. This example illustrates that the AV methods are less likely to be affected by model space constraints. In fact, in the presence of extreme multimodality, the AV methods may perform significantly better than the vanilla method whatever the range of models considered.

10. Concluding remarks

We have introduced a collection of techniques for reversible jump proposal choice, firstly in the traditional setting as introduced by Green (1995) and secondly in the more flexible saturated space setting. It was shown that several techniques used to construct proposals in Euclidean spaces (e.g. Langevin-diffusion-motivated methods and Hamiltonian AV techniques) can be extended to our setting in this paper.

The results have been applied to varied Bayesian examples, autoregressive model choice, finite mixtures and Gaussian graphical model choice. The results show that the new techniques can produce considerable improvements over (even heavily pilot-tuned) standard methods in many cases. However, the results are far from being uniformly positive towards the use of our techniques, and an important question raised by our investigation asks in what classes of problems are our methods most successful. The relative performance of our methods in comparison with vanilla techniques seems to be best in more complex problems with large numbers of models and model spaces which are highly non-linear. Thus, the zeroth-order method performs considerably better than the vanilla method in the graphical model examples, whereas the autoregressive model choice example mixes adequately using vanilla methodology, so only a marginal improvement in performance is observed with some of our methods. However, the AV methods that we introduced generally outperform vanilla methods in all the examples that we have considered.

As with Langevin algorithms, the methods proposed here could suffer from problems where the proposed variance values are totally inappropriate (as for example in the fowl bones example of Section 8). This could be caused by unrepresentative centring points, or indeed by target densities within models not being sufficiently smooth. For this reason, it will be sensible in many applications to adopt a truncation on the algorithm scaling parameters.

The approaches introduced in this paper have been largely introduced in the special case where moves to smaller dimensional models are deterministic. As we show in Section 5 (see also the examples in Ehlers and Brooks (2002) and Brooks *et al.* (2003)), there is no need for this restriction. The full generality of these approaches remains to be explored.

Additionally, within the saturated space approach, natural classes of proposals can be constructed by composing within-model candidates with deterministic between-model moves. The usefulness of this idea remains to be explored.

It is also clear that further work on the choice of centring points is necessary. This issue is only briefly touched on in this paper, where the weak non-identifiability and conditional maximization methods are described. Apart from the promising ideas introduced in Green (2002) and Ntzoufras *et al.* (2002), this is a highly undeveloped and important area.

One subsidiary point that comes out of the examples studied is the issue of convergence diagnostics for reversible jump algorithms in general. It seems that the problems that are encountered with such diagnostics in Euclidean state spaces are exacerbated in the context of reversible jump algorithms on more complex spaces. These issues are investigated further in Brooks *et al.* (2002), for example.

Acknowledgements

We are extremely grateful to Peter Green for very helpful discussions on the subject of this paper, and for allowing us access to the computer code which we modified to run the algorithms in the mixtures and graphical models examples. We are also very grateful to Ricardo Ehlers whose own work in the area of Bayesian time series and in the application of some of the methods described in this paper has provided us with valuable intuition into how and why these methods work. Finally, we are very grateful for the constructive comments made by Jon Forster, Simon Godsill, all the participants at the training and mobility of researchers workshop on reversible jump MCMC methods that was held in Spetses during August 2001 and five referees. This work has been supported by European Union training and mobility of researchers network ERB-FMRX-CT96-0095 on 'Computational and statistical methods for the analysis of spatial data' and by the Engineering and Physical Sciences Research Council under grant AF/000537.

Appendix A: State-dependent proposal density—an equivalent but alternative set-up

The notation set-up introduced in Section 2.1 and summarized in Fig. 1 can be described alternatively as follows. This is an alternative (and essentially equivalent) formulation of equation (5) which merely modifies the random seed jump distribution directly. Suppose that we are initially given a collection of random seeds drawn from a distribution function F_q. Let

$$v_{i,j,\boldsymbol{\theta}_i}(U_1, \ldots, U_{n_j - n_i}) = (F^{-1}_{1;i,j,\boldsymbol{\theta}_i}\{F_q(U_1)\}, \ldots, F^{-1}_{n_j - n_i;i,j,\boldsymbol{\theta}_i}\{F_q(U_{n_j - n_i})\}), \tag{22}$$

where $F_{l;i,j,\boldsymbol{\theta}_i}$ is the distribution function of a one-dimensional distribution, which can depend on $\boldsymbol{\theta}_i$, $1 \leqslant l \leqslant n_j - n_i$. Let $U_1, \ldots, U_{n_j - n_i}$ be a collection of state-independent random seeds drawn from distribution function F_q. If $\mathbf{U}$ is drawn from $q_{n_j - n_i}$, then $v_{i,j,\boldsymbol{\theta}_i}(\mathbf{U})$ consists of independent components $(V_1, \ldots, V_{n_j - n_i})$ with V_l having distribution function $F_{l;i,j,\boldsymbol{\theta}_i}$. In other words the algorithm now merely inputs differently distributed random variables into the canonical jump function. To apply equation (6) we need to evaluate the Jacobian term and, as discussed in Section 3, it is easily verified that

$$|J^h_{ij}(\boldsymbol{\theta}_i, \mathbf{u})| = |J^f_{ij}\{\boldsymbol{\theta}_i, v_{i,j,\boldsymbol{\theta}_i}(\mathbf{u})\}| \times \left| \frac{\partial v_{i,j,\boldsymbol{\theta}_i}(\mathbf{u})}{\partial \mathbf{u}} \right|. \tag{23}$$

The first term on the right-hand side of equation (23) is exactly that used in ordinary reversible jumps using the canonical jump functions $\{f_{i,j}\}$. Investigating the second term by using equation (22) we obtain

$$\left| \frac{\partial v_{i,j,\boldsymbol{\theta}_i}(\mathbf{u})}{\partial \mathbf{u}} \right| = \left(\prod_{l=1}^{n_j - n_i} f_{i,j,\boldsymbol{\theta}_i}[F^{-1}_{l;i,j,\boldsymbol{\theta}_i}\{F_q(U_l)\}] \right)^{-1} \prod_{l=1}^{n_j - n_i} f_q(u_q)$$

where $f_{i,j,\boldsymbol{\theta}_i}$ denotes the density corresponding to the distribution function $F_{l;i,j,\boldsymbol{\theta}_i}$. Thus, in this context we can rewrite equation (6) as

$$A_{i,j}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \frac{\pi(M_j, \boldsymbol{\theta}_j)\, r_{ji}(\boldsymbol{\theta}_j)}{\pi(M_i, \boldsymbol{\theta}_i)\, r_{ij}(\boldsymbol{\theta}_i) \prod_{l=1}^{n_j - n_i} f_{i,j,\boldsymbol{\theta}_i}(\mathbf{v})} |J_{ij}^f(\boldsymbol{\theta}_i, \mathbf{v})|,$$

where $\mathbf{v}$ is initially drawn from the distribution with independent components with respective distribution functions $F_{l;i,j,\boldsymbol{\theta}_i}$, and $\boldsymbol{\theta}_j$ is constructed from equations (5) and (22).

A.1. Example: triangular proposals

Consider triangular densities, satisfying

$$\varphi(v) = R^{-1} + \gamma v, \qquad v \in [a, b] \tag{24}$$

with centring point $v = 0$, $a \leqslant 0$ and $b \geqslant 0$. Clearly, at the centring point (corresponding to $u = 0$), the proposal density in equation (24) is independent of the value of γ. Thus, the value of R can be obtained from the zeroth-order formula (8) which will hold for all values of γ. The value of γ can then be chosen to satisfy the first-order formula (16), given this value of R. It is easy to show that

$$\gamma = \frac{1}{R} \nabla (\log[\pi_j\{M_j, f_{i,j,\boldsymbol{\theta}_i}(v)\}]\, |J_{ij}^f(\boldsymbol{\theta}_i, v)|)|_{v=0}$$

satisfies equation (16). We choose a and b so that the interval $[a, b]$ contains the centring point, i.e. $a \leqslant 0 \leqslant b$. If $2|\gamma|R^2 \geqslant 1$, then we choose one of a and b to be a location at which the proposal density becomes 0 and the other end point is chosen to ensure unit probability mass. If $2|\gamma|R^2 \leqslant 1$ this would leave the centring point outside the interval $[a, b]$. A sensible alternative is to fix the centring point to be at the midpoint of $[a, b]$, although other approaches are possible.

Appendix B: Proof of lemma 1

Consider an arbitrary set A such that $e \in A$, and $\pi(A) = a$. Then, since $e \in A$, the only way to move to a point in A^c is to move from model M_1 to model M_2. Therefore, the formula for $\kappa(A)$ in equation (17) reduces to

$$\begin{aligned}\kappa(A) &= \frac{p}{a} P(e, A^c) = \frac{p}{a} \int_{A^c} \min\left\{1, \frac{1-p}{p} \frac{f(y)}{q(y)}\right\} q(y)\, \mathrm{d}y \\ &= \frac{p}{a} \mathbf{E}_f\left[\min\left\{\frac{q(Y)}{f(Y)}, \frac{1-p}{p}\right\} \mathbf{1}_{A^c}(Y)\right] = \frac{p(1-a)}{a(1-p)} \mathbf{E}_{f|A^c}\left[\min\left\{\frac{q(Y)}{f(Y)}, \frac{1-p}{p}\right\}\right],\end{aligned} \tag{25}$$

where $f|A^c$ is just the density proportional to f restricted to A^c so that $f|A^c(y)$ equals $(1-p)\, f(y)/(1-a)$ for $\forall y \in A^c$ and 0 otherwise. (The normalization constant here is derived from the fact that $\pi(A) = p + (1-p)\, f(A)$, since $e \in A$.)

Let $\Gamma(u) = \{y : q(y)/f(y) < u\}$, and $\lambda(u) = \mathbf{P}_f\{\Gamma(U)\}$. We assume that λ is continuous. A minor modification of this argument is possible to cover the discontinuous case. Let u^* be such that $\lambda(u^*) = 1 - a$. Set $S^c = \Gamma(u^*)$. Then the expectation under f can be split into two distinct components by restricting the support first to $S^c \subseteq \Theta_2$ and $S \cap \Theta_2$ as follows:

$$\mathbf{E}_f\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right] = (1-a)\, \mathbf{E}_{f|S^c}\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right] + a\, \mathbf{E}_{f|S\cap\Theta_2}\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right].$$

Furthermore, $\min\{(1-p)/p, q(Y)/f(Y)\}$ on S^c is less than or equal to its value on $S \cap \Theta_2$. Therefore

$$\mathbf{E}_{f|S^c}\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right] \leqslant \mathbf{E}_{f|S\cap\Theta_2}\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right]$$

and so

$$\mathbf{E}_{f|S^c}\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right] \leqslant \mathbf{E}_f\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right]. \tag{26}$$

Therefore

$$\begin{aligned}
\inf_{A:\pi(A)=a,\, e\in A} \{\kappa(A)\} &\leqslant \frac{p(1-a)}{a(1-p)}\, \mathbf{E}_{f|S^c}\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right] && \text{by equation (25)} \\
&\leqslant \frac{p(1-a)}{a(1-p)}\, \mathbf{E}_{f}\left[\min\left\{\frac{1-p}{p}, \frac{q(Y)}{f(Y)}\right\}\right] && \text{by inequality (26)} \\
&\leqslant \frac{p(1-a)}{a(1-p)}\, \min\left[\frac{1-p}{p}, \mathbf{E}_{f}\left\{\frac{q(Y)}{f(Y)}\right\}\right] \\
&= \frac{p(1-a)}{a(1-p)}\, \min\left(\frac{1-p}{p}, 1\right),
\end{aligned}$$

since $\mathbf{E}_f\{q(Y)/f(Y)\} = \int q/f\ f(y)\,\mathrm{d}y = 1$. However, the last term in these series of inequalities is the value for the capacitance for any set A such that $\pi(A) = a$ and $e \in A$ in the case where $q = f$. Hence, for this case, the capacitance is maximized when $q = f$.

The argument for the case where $e \notin A$ is easy since, by reversibility, $\kappa(A) = \pi(A^c)\,\kappa(A^c)/\pi(A)$ so we can consider instead A^c which is covered by the case considered in detail above. The result therefore follows.

References

Berger, J. O. (2000) Bayesian analysis: a look at today and thoughts of tomorrow. *J. Am. Statist. Ass.*, **95**, 1269–1276.

Besag, J. (2000) Markov chain Monte Carlo for statistical inference. *Technical Report*. University of Washington, Seattle.

Brooks, S. P., Friel, N. and King, R. (2003) Model selection via simulated annealing. *J. R. Statist. Soc.* B, **65**, in the press.

Brooks, S. P. and Giudici, P. (2000) MCMC convergence assessment via two-way ANOVA. *J. Comput. Graph. Statist.*, **9**, 266–285.

Brooks, S. P., Giudici, P. and Philippe, A. (2002) On non-parametric convergence assessment for MCMC model selection. *J. Comput. Graph. Statist.*, to be published.

Brooks, S. P. and Roberts, G. O. (1999) On quantile estimation and MCMC convergence. *Biometrika*, **86**, 710–717.

Carlin, B. P. and Chib, S. (1995) Bayesian model choice via Markov chain Monte Carlo methods. *J. R. Statist. Soc.* B, **57**, 473–484.

Dellaportas, P. and Forster, J. J. (1999) Markov chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika*, **86**, 615–633.

Diaconis, P., Holmes, S. and Neal, R. M. (2000) Analysis of a non-reversible Markov chain sampler. *Ann. Appl. Probab.*, **10**, 726–752.

Duane, S., Kennedy, A., Pendleton, B. and Roweth, D. (1987) Hybrid Monte Carlo. *Phys. Rev. Lett.* B, **195**, 217–222.

Ehlers, R. S. and Brooks, S. P. (2002) Model uncertainty in integrated ARMA processes. *Technical Report*. University of Cambridge, Cambridge.

Fan, Y. and Brooks, S. P. (2000) Bayesian modelling of prehistoric corbelled tombs. *Statistician*, **49**, 339–354.

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

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

Geyer, C. J. and Møller, J. (1994) Simulation procedures and likelihood inference for spatial point processes. *Scand. J. Statist.*, **21**, 359–373.

Giudici, P. and Green, P. J. (1999) Decomposable graphical Gaussian model determination. *Biometrika*, **86**, 785–801.

Godsill, S. (2001) On the relationship between MCMC methods for model uncertainty. *J. Comput. Graph. Statist.*, **10**, 230–248.

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

———(2002) Trans-dimensional Markov chain Monte Carlo. In *Highly Structured Stochastic Systems*. Oxford: Oxford University Press. To be published.

Green, P. J. and Mira, A. (2001) Delaying rejection in Metropolis-Hastings algorithms with reversible jumps. *Biometrika*, **88**, 1035–1053.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.
Huerta, G. and West, M. (1999) Priors and component structures in autoregressive time series models. *J. R. Statist. Soc.* B, **61**, 881–899.
Jennison, C. (1997) Discussion on 'On Bayesian analysis of mixtures with an unknown number of components' (by S. Richardson and P. J. Green). *J. R. Statist. Soc.* B, **59**, 778–779.
Kendall, W. S. and Thonnes, E. (1999) Perfect simulation in stochastic geometry. *Pattn Recogn*, **32**, 1569–1586.
Lauritzen, S. L. (1996) *Graphical Models*. Oxford: Oxford University Press.
Lawler, G. and Sokal, A. (1988) Bounds on the L^2 spectrum for Markov chains and Markov processes. *Trans. Am. Math. Soc.*, **309**, 557–580.
Marinari, E. and Parisi, G. (1992) Simulated tempering. *Europhys. Lett.*, **19**, 451–458.
Møller, J. (1999) Markov chain Monte Carlo and spatial point processes. In *Stochastic Geometry: Likelihood and Computations* (eds O. E. Barndorff-Nielsen, W. S. Kendall and M. N. M. van Lieshout), pp. 141–172. London: Chapman and Hall.
Neal, R. M. (1996) *Bayesian Learning for Neural Networks*. New York: Springer.
Ntzoufras, I., Dellaportas, P. and Forster, J. J. (2002) Bayesian variable and link determination for generalised linear models. *J. Statist. Planng Inf.*, to be published.
Peskun, P. H. (1973) Optimum Monte Carlo sampling using Markov chains. *Biometrika*, **60**, 607–612.
Preston, C. J. (1977) Spatial birth-and-death processes. *Bull. Int. Statist. Inst.*, **46**, 371–391.
Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. R. Statist. Soc.* B, **59**, 731–792.
Ripley, B. D. (1977) Modelling spatial patterns (with discussion). *J. R. Statist. Soc.* B, **39**, 172–212.
Roberts, G. O., Gelman, A. and Gilks, W. R. (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.*, **7**, 110–120.
Roberts, G. O. and Rosenthal, J. S. (1998) Optimal scaling of discrete approximations to Langevin diffusions. *J. R. Statist. Soc.* B, **60**, 255–268.
Roberts, G. O. and Sahu, S. K. (1997) Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. R. Statist. Soc.* B, **59**, 291–317.
Roberts, G. O. and Tweedie, R. L. (1996) Exponential convergence of Langevin diffusions and their discrete approximations. *Bernoulli*, **2**, 341–363.
Sargent, D. J., Hodges, J. S. and Carlin, B. P. (2000) Structured Markov chain Monte Carlo. *J. Comput. Graph. Statist.*, **9**, 217–234.
Stephens, M. (2000) Bayesian analysis of mixture models with an unknown number of components: an alternative to reversible jump methods. *Ann. Statist.*, **28**, 40–74.
Tierney, L. (1998) A note on the Metropolis Hastings algorithm for general state spaces. *Ann. Appl. Probab.*, **8**, 1–9.
Troughton, P. T. and Godsill, S. J. (2001) MCMC methods for restoration of nonlinearly distorted autoregressive signals. *Signal Process.*, **81**, 83–97.
Whittaker, J. (1990) *Graphical Models in Applied Multivariate Statistics*. Chichester: Wiley.

Discussion on the paper by Brooks, Giudici and Roberts

Christian P. Robert (*Centre de Recherche en Economie et Statistique and Université Dauphine, Paris*)

This paper aims to develop general strategies for improving jumps between models in reversible jump Markov chain Monte Carlo (MCMC) algorithms, which is quite an important and timely goal. Indeed, in practical implementations of the method, we usually find that the choice of proposals is paramount: in many cases, the 'natural choice' leads to a zero acceptance probability and the construction of well-tuned moves is often quite costly. Given that the reversible jump MCMC method is an essential part of the Bayesian toolbox, at least in Bayesian exploratory analysis, a debate is needed for more global strategies on the choice of proposals.

The first appealing feature, at the core of the paper, is that image parameters that give a Metropolis–Hastings probability of *1* should be identified as *pivotal quantities*, just like the current value is a pivot for the random-walk Metropolis–Hastings move. The authors then propose 'higher order' methods where some derivatives of the probability are set to 0, but I find this less appealing, because it considerably adds to the complexity of the algorithm.

Obviously, the authors mostly focus on *scale*, rather than *location–scale* tuning. The choice of the centring function $c_{i,j}(\theta_i)$ is not discussed much further in the paper, even for the case of nested models. For instance, the resolution of

$$L_i(\text{data}|\theta_i) = L_j[\text{data}|f_{i,j}\{\theta_i, b_{i,j}(\theta_i)\}],$$

suggested in Section 2.2, is usually intractable. Moreover, by considering only the likelihood, it may provide values with very small prior probabilities. In addition, the moves between *non-nested models* are not necessarily natural. Take for instance a set of generalized linear models M_{ij} including some distributions f_i and link functions l_j such that $y|x \sim f_i\{y|l_j(x^{\mathrm{T}}\beta_{ij})\}$; the parameter β_{ij} depends on the choice of both f_i and l_j, and the move from model M_{ij} to model M_{uv} should not be centred at β_{ij}. At the very least, we must have an invariant over the models like some moments of y or the likelihood, but the resolution of the corresponding equations is likely to be costly. Similarly, weak non-identifiability does not work well in this non-nested example.

This is an opportunity to stress a general *statistical* problem with reversible jump methods, which is the systematic recourse to the *same* parameters when jumping between nested models (Section 2.1.1). Most researchers on model choice state that, on the contrary, the interpretation of parameters *should* change between models. For instance, when considering an autoregressive AR(p) and an AR($p+1$) model, the meaning of a_1 is not the same in both models. In this set-up, under stationarity, the correct parameterization is either through *partial autocorrelations*, which are quite unrelated for orders p and $p+1$, or through the (inverse) roots λ^p of the lag polynomial

$$\prod_{i=1}^{p} (1 - \lambda_i B) X_t = \varepsilon_t, \tag{27}$$

as in Huerta and West (2000), which, again, do not remain similar between models and thus make the 'natural' centring

$$c_{p,\,p+1}(\lambda^p) = (\lambda^p, 0) \tag{28}$$

questionable. Under uniform priors for the real and complex roots λ_j,

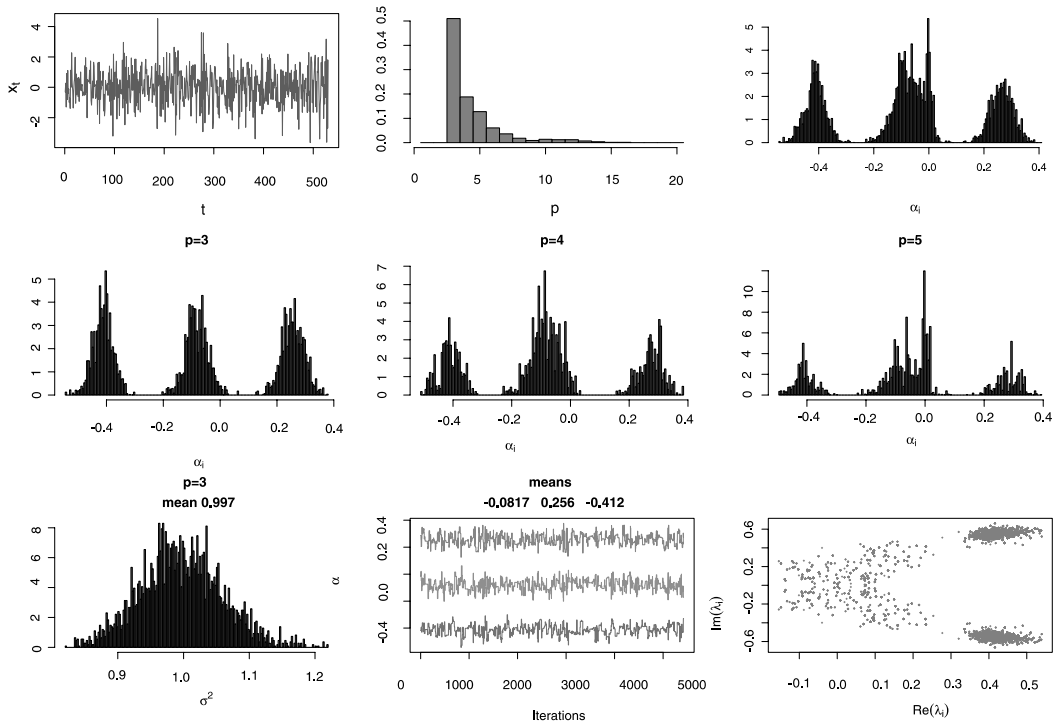

Fig. 5. Graphical representation of the performances of a reversible jump algorithm for the root parameterization of the AR(p) model, based on a simulated data set of 530 points (upper left) with true parameters α_j ($-0.1, 0.3, -0.4$) and $\sigma = 1$: the first histogram gives the posterior distribution of p and the following histograms the distribution of the natural parameters α_j, for different values of p, and of σ^2; the final graph is a scatterplot of the complex roots of the lag polynomial and the bottom middle graph shows the evolution of α_1, α_2 and α_3

$$\frac{1}{\lfloor k/2 \rfloor + 1} \prod_{\lambda_i \in \mathbb{R}} \frac{1}{2}^{\mathbb{I}_{|\lambda_i|<1}} \prod_{\lambda_i \notin \mathbb{R}} \frac{1}{\pi}^{\mathbb{I}_{|\lambda_i|<1}}$$

and proposals based on the priors, a standard reversible jump algorithm does not encounter convergence problems (Fig. 5). None-the-less, if the uniform prior is replaced with a beta-type proposal,

$$\varphi(\lambda_i) \propto |\lambda_i|^{\beta}, \qquad \beta > 0,$$

where β is to be optimized, the centring equation (8) is not defined when equation (28) is used. (This points out a common measure theoretic difficulty with criteria which depend on the value of a density at a specific point and suggests a more *integrated* alternative for the centring equation.)

Another very appealing feature of the paper is to consider in parallel a kind of saturated model, called the *dual space approach*, where the parameter to be simulated is of constant dimension $n_{\max}$ by incorporation of the remaining auxiliary variables, a feature reminiscent of Carlin and Chib (1995) and Godsill (2001). I find this representation of reversible techniques very natural. At a deeper level, it seems to me that this could lead to more general (including non-reversible) MCMC algorithms in variable dimension models, since moves between models are then Gibbs steps.

The idea of using the $u_{i,r}$s as auxiliary variables is particularly exciting, in that it eliminates the apparent difficulty of moving between spaces of different dimensions. It also clarifies what happens to the $u_{i,j}$s when the move is not between M_i and M_j. It, however, presupposes some degree of homogeneity between models in that the $u_{i,r}$s remain (almost) the same between moves, i.e. some generic feature in the $u_{i,r}$s, like uniformity, and thus a more elaborate construction of the transforms $f_{i,j}$. This is particularly true for (dimension-) correlated proposals. For instance, although the correlated auxiliary variable methods are performing better than the 'vanilla' algorithm in the mixture example, it is quite difficult to fathom why this is so and this seems to indicate that the methods proposed require harder tuning than alluded in the paper. (The comparison between the three methods in Section 9 is definitely unfair, in that the truncation on $k \leqslant 5$ seems to invalidate the vanilla algorithm, whereas, in Richardson and Green (1997), mixing was quite satisfactory. As shown in Fig. 6, using a C program of my own, the truncation of k at $k \leqslant 5$ does not appear to hamper convergence in such a dramatic way.)

The paper offers a well-thought-out reflection on reversible jump techniques, in particular by rephrasing the moves in terms of a centre in the arrival space, contains path breaking innovations in designing reversible jump proposals and opens new and broad avenues for designing reversible jump algorithms, while

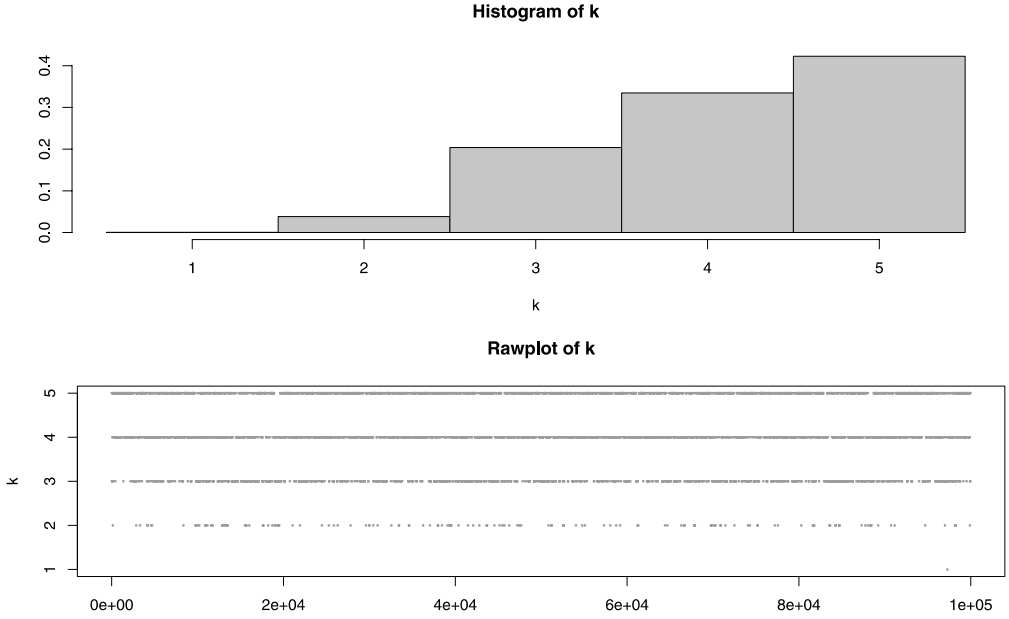

Fig. 6. Histogram and raw data plot of 100 000 k produced by Richardson and Green's (1997) algorithm for the enzyme data set, under the imposed constraint $k \leqslant 5$

calling for further research in producing generic proposals. I thus have great pleasure in proposing the vote of thanks!

Xiao-Li Meng (*Harvard University, Cambridge, and University of Chicago*)
Initially I was a little puzzled at being invited to discuss this paper, as I could not locate my name anywhere in it. Then I realized that this perhaps was intentional, for the second discussant is often perceived to be the 'bad guy'—an 'uncited' discussant tends to be more critical. As a 'good guy', one often gets away with saying 'The authors are to be congratulated for a very interesting paper. Now look at what I have done.' Being a bad guy, however, one actually has to read the paper. I read every word (at least through the first page), partially in the hope of finding flaws to be labelled 'rubbish', a word, as I understood which was previously used in such a context. Unfortunately for me as the second discussant, but fortunately for readers, the only place where I could possibly do so is at the third sentence of the paper. This sentence, if read literally, suggests that there is a *Bayesian model determination* problem beyond the *choice of prior* and the *specification of the likelihood*. What could that be? Of course it is unfair to blame the authors for this common misuse of 'model determination', which should be 'submodel determination'. But even with this qualification there remains an inconsistency. Few would label inference regarding a continuous parameter (e.g. a normal mean) a submodel determination problem, even though paradigmatically it is not different from making inference about, say, the order of an autoregressive model. Perhaps because of my recent involvement in a joint paper (to be read to the Royal Statistical Society in December 2002) that largely centres on the meaning of 'unknown' in likelihood or Bayesian modelling, I am a little more sensitive to semantic issues that might have unintended consequences.

Having done my duty as the bad guy, let me return to my normal role as a good guy so that I can justifiably advertise some of my own work. First, in van Dyk and Meng (2001), we showed mathematically that the auxiliary variable (AV) and data augmentation (DA) methods are trivially equivalent, although they were proposed for different purposes. This connection further emphasizes that efficient construction of AVs, just as with DA, is a matter of art, a fact that is clearly demonstrated by the current paper. More importantly, the AV and DA literatures have much to share. For example, the saturated space approach is commonly used with DA for turning irregular problems into regular ones. As another example, conditional augmentation (Meng and van Dyk, 1999) employs the same strategy as conditional maximization, i.e. both define a class of candidate DA–AV schemes and then optimize according to a specified criterion.

In Meng and Schilling (2002), we investigated some strategies to match densities via warping their geometric shapes. Warp I resembles the zeroth-order approach, as both attempt to match a 'centre'. Similarly, both warp II and the first-order method aim to match a measure of 'spread'. The geometric approach that we took might also complement the authors' algebraic extensions to higher orders. Our warp III strategy symmetrizes any density via group averaging, such as rotation and reflection. As an illustration of the possibility of a *transdimensional* warp III, consider moving from an arbitrary density $p_+(\tilde{X})$ on R^1_+ to an elliptically symmetric density on R^2. This can easily be done by revolving p_+ around the Z-axis in R^3 (with the appropriate Jacobian). Or, statistically, let $\phi \sim \text{uniform}(0, 2\pi)$ be an AV, independent of $\tilde{X}$, and then let $X = \sigma\tilde{X}\cos(\phi)$ and $Y = \sigma\tilde{X}\sin(\phi)$, where σ can be determined, for example, by 'optimal scaling'. The 'reversed jump' is simply $\tilde{X} = \sqrt{(X^2 + Y^2)}/\sigma$. My emphasis here is not on the simple polar transformation but on the fact that probability masses for (elliptically) symmetric densities tend to be more evenly distributed, and thus they might be more suitable as proposal densities in some problems. Of course, this is speculative, but my positive experiences with warp III gives me some hope that transdimensional warping is not science fiction.

To echo my initial 'complaint' about the lack of citation, let me end my discussion with a citation ranking in *Science Watch* (May–June 2002). Among the top 25 ranks of *mathematicians* worldwide in terms of papers published and cited during 1991–2000, statisticians occupied 19 spots! Parts of the list reads like a 'who's who?' for the Markov chain Monte Carlo club, including two of the Society's Presidents, Adrian Smith (number 3) and Peter Green (number 13). I certainly hope that this paper will help to increase Gareth Roberts's current ranking (16), and, with a little help from the authors, my own ranking would be moved up as well!

With that thought, it gives me great pleasure to second the vote of thanks for this potentially highly cited paper!

The vote of thanks was passed by acclamation.

Jesper Møller (*Aalborg University*)
I enjoyed reading this stimulating paper. Section 1.1 mentions briefly the many modifications, extensions

and variations of reversible jump methodology, in particular approaches based on point and birth–death processes. As illustrated below I find this a promising direction of research, where many points need further development, not least in connection with Bayesian model determination applications.

One example is provided by Stephens (2000); cf. Section 1.1. See also Cappé *et al.* (2002). Another, simpler, example concerns a setting similar to that of Section 1.2, with a target density $\pi(\theta_k) > 0$ for $k = 1, \ldots, k_{\max}$ and $\theta_k = (a_1, \ldots, a_k) \in \mathbb{R}^k$. As a simple alternative to the Metropolis–Hastings moves in the paper, consider a birth–death process $X = (X_t)_{t \geqslant 0}$ with similar types of move, i.e. a continuous time Markov chain with

(a) rate $\beta(\theta_k)\, b(\theta_k, a_{k+1})$ for a birth $\theta_k = (a_1, \ldots, a_k) \to \theta_{k+1} = (a_1, \ldots, a_{k+1})$, where $\beta(\theta_k) > 0$ and $b(\theta_k, \cdot)$ is a density function for $k = 1, \ldots, k_{\max} - 1$, and
(b) rate $d(\theta_{k+1}) > 0$ for a death $\theta_{k+1} = (a_1, \ldots, a_{k+1}) \to \theta_k = (a_1, \ldots, a_k)$.

Then reversibility is ensured by detailed balance,

$$\pi(\theta_k)\, \beta(\theta_k)\, b(\theta_k, a_{k+1}) = \pi(\theta_{k+1})\, d(\theta_{k+1}) \qquad \text{for } k = 1, \ldots, k_{\max} - 1. \tag{29}$$

Assuming this, knowing the birth-rate we know the death-rate (and vice versa), but how do we choose the birth-rate? A computationally simple but naïve strategy is to let $\beta(\theta_k)\, b(\theta_k, \cdot)$ depend on k only (in a Bayesian setting the birth-rate may depend on the data). The opposite strategy is to let $d(\theta_{k+1}) = d_{k+1}$ depend on k only (a similar strategy is often used for point processes; see for example Ripley (1977), Baddeley and Møller (1989) and Kendall and Møller (2000)). For the particular target density in Section 1.2, if $d(\theta_{k+1}) \equiv 1$, condition (29) implies that, for $k = 1, \ldots, k_{\max} - 1$,

$$\beta(\theta_k) = \exp\{c_2(\theta_k)^2 / 2\, c_1(\theta_k)\},$$

$$b(\theta_k, \cdot) \sim N\{c_2(\theta_k)/c_1(\theta_k),\, 1/c_1(\theta_k)\},$$

where

$$c_1(\theta_k) = \frac{1}{\sigma_a^2} + \frac{1}{\sigma_\varepsilon^2} \sum_{t=k_{\max}+1}^{T} x_{t-(k+1)}^2,$$

$$c_2(\theta_k) = \frac{1}{\sigma_\varepsilon^2} \sum_{t=k_{\max}+1}^{T} \left(x_t - \sum_{\tau=1}^{k} a_\tau x_{t-\tau} \right) x_{t-(k+1)}.$$

Thus it is straightforward to make simulations. For other types of target densities it may be less straightforward.

The construction of a birth–death process above can easily be modified if we do not limit the number of components $k = 1, 2, \ldots$. For details see `http://www.math.auc.dk/~jm/discussionBrooks-etal.ps`; where how the coupling method in Kendall and Møller (2000) may be modified and applied to perfect simulation is also discussed. Incidentally, the ideas in Møller and Nicholls (1999) and Brooks *et al.* (2002) for making perfect simulated tempering simulations also apply in connection with reversible jump Markov chain Monte Carlo algorithms.

It is tempting to extend the approach of birth–death processes to general reversible jump processes, i.e., in the context of model selection problems, continuous time Markov chains which jumps within and between the spaces (M_i, Θ_i); see for example Cappé *et al.* (2002). One particular problem is to develop strategies for finding reasonable jump rates. The paper by Brooks, Giudici and Roberts might serve as an inspiration for developing such strategies.

Jeffrey S. Rosenthal (*University of Toronto*)

Statistical models of varying dimension are becoming increasingly important for a wide variety of applications. Successful use of such models, especially in a Bayesian context, requires sophisticated Markov chain Monte Carlo (MCMC) algorithms for exploring transdimensional distributions, and the authors are to be congratulated for focusing on this topic.

It is desirable where possible to prove theoretical convergence rate bounds on MCMC algorithms (see for example Rosenthal (1995)), and this poses a particular challenge for transdimensional chains. Inspired by the current paper, we have analysed in detail the convergence rate and properties of the simple transdimensional chain described in lemma 1 therein. Of particular interest is the case when $p \approx \frac{1}{2}$, and $q \approx f$. The chain is then nearly *periodic*, in that it jumps repeatedly between the two different dimensions at

each iteration. In that case the law of θ_k does *not* rapidly approach stationarity, even though the chain is exploring the state space well.

A solution is to consider θ_{B_k} in place of θ_k, where $B_k \sim \text{binomial}(2k, \frac{1}{2}) \approx k$ is independent of the chain itself (so that θ_{B_k} corresponds to running $2k$ iterations of a modified chain which half the time does nothing). In Rosenthal (2002a, b) we prove rapid convergence for this modified chain, with extensive generalizations to other 'sampled chains' of the form

$$P^{\mu} = \sum_n \mu\{n\} P^n.$$

We hope that the future brings many more interactions between transdimensional MCMC algorithms on the one hand and theoretical Markov chain analysis on the other.

C. Jennison and M. A. Hurn (*University of Bath*) **and F. Al-Awadhi** (*Kuwait University*)

We have tackled the problem of identifying cells in a confocal microscope image by using a marked point process of elliptical non-overlapping cells as the prior image model. In sampling from the posterior distribution, new cells are created in 'birth' moves and removed in 'death' moves; other moves allow cells to be split or merged. In adding a new cell, six parameter values are generated for its location, orientation, size and intensity. Even with a carefully chosen proposal distribution, in six-dimensional space acceptance probabilities can be vanishingly small with tens of thousands of proposals required for each acceptance. These difficulties are of a different order from those in the authors' examples and, since successful jumps are rare, little information is available to 'tune' proposal distributions adaptively.

Our solution is to process each jump proposal, creating a more plausible sample from the posterior distribution before considering acceptance. For simplicity, we discuss only birth and death moves here. Following the authors' notation, let $\pi(M_i, \theta_i)$ denote the subdensity with respect to Lebesgue measure on R^{6i} of that part of the posterior distribution with i cells present. If the current image θ_i contains i cells a birth move is chosen with probability $r_{i,i+1}$ and a new cell generated to give state θ'_{i+1} from density $q_{i,i+1}(\theta_i, \theta'_{i+1})$. The new cell's parameters are then updated by k transitions of a Markov chain Monte Carlo sampler with detailed balance with respect to a density $\pi^*(M_{i+1}, \theta_{i+1})$ on $R^{6(i+1)}$, leading to the final proposal θ^*_{i+1}. Let P denote the transition kernel for the full sequence of k moves from θ'_{i+1} to θ^*_{i+1} so, by detailed balance,

$$\pi^*(M_{i+1}, \theta'_{i+1})\ P(\theta'_{i+1}, \theta^*_{i+1}) = \pi^*(M_{i+1}, \theta^*_{i+1})\ P(\theta^*_{i+1}, \theta'_{i+1}).$$

In the reverse death move, chosen with probability $r_{i+1,i}$, a cell is selected for deletion, updated k times under the sampler for $\pi^*(M_{i+1}, \theta_{i+1})$ and then deleted. The birth move's acceptance probability is

$$\alpha_{i,i+1}\{(M_i, \theta_i), (M_{i+1}, \theta^*_{i+1})\} = \min\{1, A_{i,i+1}(\theta_i, \theta^*_{i+1})\},$$

where

$$\begin{aligned} A_{i,i+1}(\theta_i, \theta^*_{i+1}) &= \frac{\pi(M_{i+1}, \theta^*_{i+1}) r_{i+1,i}\ \{1/(i+1)\}\ P(\theta^*_{i+1}, \theta'_{i+1})}{\pi(M_i, \theta_i) r_{i,i+1}\ q_{i,i+1}(\theta_i, \theta'_{i+1})\ P(\theta'_{i+1}, \theta^*_{i+1})} \\ &= \frac{\pi(M_{i+1}, \theta^*_{i+1}) r_{i+1,i}\ \{1/(i+1)\}\ \pi^*(M_{i+1}, \theta'_{i+1})}{\pi(M_i, \theta_i) r_{i,i+1}\ q_{i,i+1}(\theta_i, \theta'_{i+1})\ \pi^*(M_{i+1}, \theta^*_{i+1})}. \end{aligned} \tag{30}$$

Without the k transitions under the sampler for $\pi^*(M_{i+1}, \theta_{i+1})$, we would have

$$A_{i,i+1}(\theta_i, \theta'_{i+1}) = \frac{\pi(M_{i+1}, \theta'_{i+1}) r_{i+1,i}\ \{1/(i+1)\}}{\pi(M_i, \theta_i) r_{i,i+1}\ q_{i,i+1}(\theta_i, \theta'_{i+1})}, \tag{31}$$

in which

$$\pi(M_{i+1}, \theta'_{i+1}) / q_{i,i+1}(\theta_i, \theta'_{i+1}) \tag{32}$$

is likely to be very small when $q_{i,i+1}(\theta_i, \theta'_{i+1})$ is not well matched to $\pi(M_{i+1}, \theta_{i+1})$. Making k transitions with detailed balance with respect to π does not help as equation (30) reduces to equation (31) for $\pi^*(M_{i+1}, \theta_{i+1}) \propto \pi(M_{i+1}, \theta_{i+1})$. Improved acceptance rates can be achieved by defining $\pi^*(M_{i+1}, \theta_{i+1})$ as a distribution intermediate between $q_{i,i+1}(\theta_i, \theta_{i+1})$ and $\pi(M_{i+1}, \theta_{i+1})$, so the one very small term (32) is replaced by the product of two moderately small terms

$$\pi(M_{i+1}, \theta^*_{i+1})/\pi^*(M_{i+1}, \theta^*_{i+1})$$

and

$$\pi^*(M_{i+1}, \theta'_{i+1})/q_{i,i+1}(\theta_i, \theta'_{i+1}).$$

The analysis of a simple normal example in Al-Awadhi *et al.* (2002) shows that this method can improve low acceptance rates dramatically. Success in our object recognition problem is achieved with $\pi^*(M_{i+1}, \theta_{i+1})$ defined as a modification of the posterior distribution in which the likelihood term is 'tempered' by raising it to a power less than 1.

Peter McCullagh (*University of Chicago*)
Most models that I encounter in applications are linear, such as factorial models, or generalized linear, such as log-linear models for contingency tables. It is invariably the case that the set of models under consideration constitutes a lattice of subspaces or subrepresentations, closed under intersection and vector spans.

This paper seems to deal with models as unrelated vector spaces making no explicit use of any embeddings that may exist. But, when we have a lattice of models and submodels, there is a natural insertion from each subspace into its parents. Further, if the model spaces are regarded as inner product spaces, there is a natural projection in the reverse direction.

I wonder whether any simplification might follow from such lattice structures.

The following contributions were received in writing after the meeting.

Christophe Andrieu (*University of Bristol*) **and Arnaud Doucet** (*University of Cambridge*)
We comment on the introduction of auxiliary variables, and in particular their use as a way of learning about the target distribution. The techniques described are borrowed from automatic control and might be useful in automating the design of reversible jump Markov chain Monte Carlo (RJMCMC) algorithms. We start with a simple motivating example from Green (2002). In the context described by Green (2002) it is suggested to use the mean μ_k and covariance Σ_k of the target distributions to define efficient jump transformations. As the mean and covariance are unknown, they are estimated by using some pilot runs. However, the recursions (here iteration $i + 1$)

$$\mu_k(i+1) = (1 - \gamma_{i+1})\, \mu_k(i) + \gamma_{i+1}\, x(i+1),$$
$$M_k(i+1) = (1 - \gamma_{i+1})\, M_k(i) + \gamma_{i+1}\, x(i+1)\, x(i+1)^{\mathrm{T}}$$

could also be used to estimate the quantities of interest, which are then fed back in the sampler; $\{\gamma_i\}$ is a decreasing step size sequence and $\{x(i)\}$ is the output of our 'MCMC' algorithm. These equations are connected to the autoregressive technique advocated by the authors but present clear different ergodic properties. Our recursion is very close to a method proposed in Haario *et al.* (2001), and it is a particular case of a much more general framework.

Assume that the MCMC algorithm depends on a tuning parameter β. In the context of the Metropolis algorithm, β could be the variance of the proposal distribution. We define a cost function that characterizes the statistical properties of the chain, of the form

$$h(\beta) = \int H(\beta, w)\, \nu_\beta(\mathrm{d}w),$$

for some function H and a probability measure ν_β to be defined later.

Assume that the optimal value for β is such that $h(\beta) = 0$. An algorithm to find the solution to this equation is Robbins–Monro recursion (Robbins and Monro, 1951), a special instance of which is given by

$$\beta_{i+1} = \beta_i + \gamma_{i+1}\, H(\beta_i, w_{i+1}),$$
$$w_{i+1} | (\theta_0, w_1, \ldots, w_i) \sim P_{\beta_i}(w_i, \mathrm{d}w_{i+1}),$$

where P_β is a kernel which admits ν_β as an invariant distribution. This is a noisy gradient algorithm. In Green's example, $w = x$,

$$H(\beta, w) = \begin{pmatrix} x \\ xx^{\mathrm{T}} \end{pmatrix} - \beta,$$
$$\nu_\beta(\mathrm{d}w) = \pi(\mathrm{d}x).$$

Adjusting the expected acceptance probability of the sampler, minimizing the autocorrelation time can be formulated in this way, for various Hs and μ_θs (Andrieu and Robert, 2001, 2002). For the expected acceptance probability

$$h(\beta) = \int_{X^2} \left\{ 1 \wedge \frac{\pi(\mathrm{d}y)\, q_\beta(y, \mathrm{d}x)}{\pi(\mathrm{d}x)\, q_\beta(x, \mathrm{d}y)} - \alpha_* \right\} \pi(\mathrm{d}x)\, q_\beta(x, \mathrm{d}y)$$

where α_* is the target acceptance rate. Here $w = (x, y)$

$$H(\beta, w) = 1 \wedge \frac{\pi(\mathrm{d}y)\, q_\beta(y, \mathrm{d}x)}{\pi(\mathrm{d}x)\, q_\beta(x, \mathrm{d}y)} - \alpha_*,$$

and $\nu_\beta(\mathrm{d}w) = \pi(\mathrm{d}x)\, q_\beta(x, \mathrm{d}y)$.

In the context of RJMCMC sampling, there are numerous potential applications of this idea such as parameterizing the jump function f_β or v_β and then optimizing some criterion.

Naturally the chain $\{x(i)\}$ is no longer Markov, and we might question the ergodicity properties of such chains. Precise results have been derived in Andrieu and Moulines (2002) that give useful quantitative bounds on the convergence of ergodic averages calculated from such chains.

Finally, a question of interest is that of improving the mixing properties of the sampler. One can suggest minimizing the first return time to a given small set C associated with a probability measure λ,

$$\tau_C = \mathbb{E}_\lambda \left\{ \sum_{i=1}^{\infty} \mathbb{I}_{C^c}(X_i) \right\},$$

which does not depend on π.

Petros Dellaportas and Ioulia Papageorgiou (*Athens University of Economics and Business*)
The authors are to be congratulated for making advances in an important methodological tool for Bayesian model determination.

The power of reversible jump sampling, compared with other existing Markov chain Monte Carlo model determination methods, is its ability to search in model space rather than approximating Bayes factors. This model searching is achieved by employing either *local* or *global* moves in the model space (Dellaportas *et al.*, 2002), but clearly the latter are more difficult since the relationship between the parameters of the models is less obvious. We would like to contribute in the discussion of Section 2 by pointing out that there are a series of tricks that may be employed to achieve efficient jumps between models.

Assume that the finite mixtures of normals problem tackled by Richardson and Green (1997) needs to be extended to the multivariate normal distributions setting. Clearly, the moment matching approach is not the solution in this case, since an immediate requirement is to propose covariance matrix elements that preserve the positive definiteness. A solution suggested by Dellaportas and Papageorgiou (2002) is the following. First, within each model, employ the usual parameterization that leads to the usual conjugate Gibbs algorithm; see, for example, Dellaportas (1998). Second, before applying f, perform a reparameterization $\boldsymbol{\theta}_i \rightarrow \boldsymbol{\phi}_i \in \boldsymbol{\Phi}$ and then search for good jumps operating on $\boldsymbol{\Phi}$. In this set-up, the parameterizations that are appropriate are, of course, those that release the need to impose the positive definiteness restriction on the covariance matrices, and can be viewed as part of f. For more details see Dellaportas and Papageorgiou (2002).

Ricardo S. Ehlers (*Universidade Federal do Paraná*)
The results in this paper are of great importance in the practical implementation of reversible jump Markov chain Monte Carlo algorithms and the applied user will certainly benefit from it.

The authors focus on a specific implementation of the reversible jump sampler where new parameters are generated and existing ones remain fixed. In the context of autoregressive models, the expressions for the proposal parameters obtained via the second-order method correspond to the posterior conditional distribution of a_{k+1} given $a_1, \ldots, a_k$ under the higher order model. Thus, the posterior conditional is chosen as the best proposal distribution for the new parameter. It is worth noting that the same solution is obtained whatever centring point is chosen since $\mathbf{u}$ drops out of the two simultaneous equations to be

solved. It would be interesting to investigate whether there is a general result for this, i.e. under what circumstances does the centring point drop out?

These methods may be extended to the case where models are still nested, but reverse moves are no longer deterministic as some of the existing parameters are changed as part of the jump. For this type of move, the acceptance ratio needs to be altered to include a proposal density in both the numerator and the denominator. In the autoregression example, we can propose a move from a model of order k with coefficients $\mathbf{a} = (a_1, \ldots, a_k)$ to a model of order k' with coefficients $\mathbf{a}' = (a'_1, \ldots, a'_{k'})$ by generating new values for the whole vector of autoregressive coefficients directly in the k'-dimensional space (keeping the error variance fixed). In terms of dimension matching, this is equivalent to generating a k'-dimensional random vector $\mathbf{u} = (u_1, \ldots, u'_k)$ from a proposal distribution $q(\mathbf{u})$ and then setting the change of variables as $\mathbf{a}' = \mathbf{u}$ and $\mathbf{u}' = \mathbf{a}$, i.e. $f(\mathbf{a}, \mathbf{u}) = (\mathbf{u}, \mathbf{a})$ which has unit Jacobian. Extending the kth-order method to this case by simply taking the derivatives of A with respect to both $\mathbf{a}$ and $\mathbf{u}$, the second-order method again suggests the posterior conditional distributions but this time conditioning on σ_ε^2 alone; see Ehlers and Brooks (2002a).

Elena A. Erosheva and Stephen E. Fienberg (*Carnegie Mellon University, Pittsburgh*)

We congratulate the authors on their stimulating paper. We hope that they can suggest how to adapt their approach to a problem with which we have been struggling.

We are interested in what is known as the grade-of-membership model (see Manton *et al.* (1994) and Erosheva *et al.* (2002)). For a random sample of subjects, we observe J dichotomous responses $x_1, \ldots, x_J$. We assume that there are K basis subpopulations, which are determined by the conditional (positive) response probabilities, λ_{kj}, $j = 1, \ldots, J$. The subjects are characterized by their degrees of membership in each of the subpopulations $g = (g_1, \ldots, g_K)$, which are non-negative and add to 1. Conditionally on the subject's membership scores g, the subject's response probability for item j is given by a convex combination

$$\Pr(x_j = 1|g) = \sum_k g_k \lambda_{kj}.$$

We assume that the responses $x_1, \ldots, x_J$ are conditionally independent, given the membership scores, and that the membership scores g have a Dirichlet distribution with parameters $\alpha = (\alpha_1, \ldots, \alpha_K)$. By using a data augmentation procedure, we obtain a posterior distribution of the parameters via a Metropolis–Hastings-within-Gibbs algorithm. The current implementation of the algorithm involves separate Metropolis–Hastings steps for the hyperparameters, reparameterized as $\alpha_0 = \Sigma_k\, \alpha_k$ and $\xi = \alpha/\alpha_0$, and a Gibbs sampler for the structural parameters, $\boldsymbol{\lambda} = \{\lambda_{kj} : k = 1, \ldots, K;\ j = 1, \ldots, J\}$, membership scores g and the variables from data augmentation (Erosheva, 2002).

To date we have fitted the model to a 2^{16}-table, separately for $K = 2, 3, 4, 5$. Incrementing the number of subpopulations from K to $K + 1$ produces 17 additional structural parameters, $\lambda_{K+1,\,j}$, $j = 1, \ldots, 16$, and α_{K+1}, and also increases the number of incidental parameters linearly with the number of subjects. As we have increased K, we have observed a slow-down in mixing, especially for the hyperparameter α_0, and the Markov chain Monte Carlo algorithm takes longer to achieve convergence.

The grade-of-membership model is a generalized mixture model which assumes partial instead of complete membership in a component. This gives us more incidental parameters. Since there is already poor mixing for the hyperparameters of the model, the question is whether the chain with a reversible jump can achieve reasonable mixing properties. Do the authors have any advice on how to adapt their approach in this circumstance?

We are interested in various related applications of essentially the same structure but where the number of variables and the number of subpopulations are considerably larger (and where the speed of convergence is also important). Is there really any hope for reversible jump sampling in this context?

Jonathan J. Forster and Roger C. Gill (*University of Southampton*)

This paper provides welcome insight into the difficult problem of how to choose proposal distributions in transdimensional Markov chain Monte Carlo sampling. We comment on one area which the authors did not discuss in detail. In Section 2, they formulate the problem of proposal construction given a specified jump function f. The choice of jump function can be critical, and the obvious choice can fail badly.

Consider moves between nested models, where the 'down-dimension' moves are taken to be deterministic. Then, the first- and higher order methods (taking all derivatives of $\log(A)$) correspond to matching derivatives of the log-proposal and log-conditional posterior density functions at a centring point. This

can lead to appealing proposals such as a normal, centred at the posterior conditional mode, with variance given by the negative inverse second derivative of the log-conditional posterior density at the mode. However, whether such a proposal generates a parameter value with non-negligible posterior probability under the proposed model depends critically on the jump function.

Consider the 'pines' data used by Carlin and Chib (1995) and Dellaportas *et al.* (2002) to illustrate Bayesian model determination based on Markov chain Monte Carlo sampling. There are two competing non-nested models: $1+X$ and $1+Z$. Here, we augment the model space to include the model $1+X+Z$ and only allow transitions between nested models. As the authors suggest, the obvious jump function for two nested models is the identity function, preserving values of coefficients common to both models. However, this fails when the parameters are highly dependent in the larger model, as here. Despite using the optimal proposal location and scale, it is impossible to generate a successful transition between models $1+X+Z$ and $1+X$. Hence this approach fails to determine the relative probabilities of $1+X$ and $1+Z$.

An alternative approach, appropriate in any generalized linear model determination problem, takes the proposed value of the linear predictor in a down-dimension move to be the orthogonal projection of the current linear predictor (an element of V_0 say) onto the subspace $V_1 \subset V_0$ defined by the proposed model (orthogonal with respect to an estimated posterior covariance inner product). The corresponding up-dimension move then proposes the linear predictor as the sum of the current value and a proposal generated in $V_0 \cap V_1^{\perp}$ (**u** parameterizes this space). A by-product of orthogonality is that the optimal proposal for **u**, based on the first- and second-order methods, is independent of the current parameter values. For the pines data this approach leads to a mobile chain, with accepted jumps on around 40% of proposals, and an accurate estimate of the relative probabilities of $1+X$ and $1+Z$ within a few thousand iterations.

We are currently investigating this approach on more testing examples, including autoregressive processes as in the current paper.

Nial Friel (*University of Glasgow*)

The authors have provided a methodology for tackling the often troublesome problem of choosing proposal distributions for moves to differing dimensional spaces. As they point out, the difficulty in choosing such proposal distributions may partly explain why the vast potential of reversible jump Markov chain Monte Carlo (MCMC) methods has, to a certain extent, not yet been fully realized. A further reason may be that, until now, reversible jump MCMC sampling has remained solely in a Bayesian context. It is possible, however, to apply this methodology in a classical setting.

Classical model selection is often based on finding model parameters that maximize the likelihood function, typically with the addition of a penalty term, e.g. the Akaike information criterion AIC. Such problems can be tackled by combining reversible jump MCMC sampling within a simulated annealing framework (Brooks *et al.*, 2003). The idea is to embed the objective function $f(\boldsymbol{\theta}_k, m_k)$ that we wish to optimize over models m_k with parameters $\boldsymbol{\theta}_k$, in the Boltzmann distribution defined by

$$b_T(\boldsymbol{\theta}_k, m_k) \propto \exp\left\{-\int (\boldsymbol{\theta}_k, m_k)/T(t)\right\}.$$

Here $T(t)$ denotes a temperature schedule, defined such that $T(t) \to 0$ slowly, as $t \to \infty$. The simulated annealing algorithm may then be adapted by introducing reversible jump MCMC moves so that for each temperature we move not only within but also between models. We term this set-up *transdimensional simulated annealing* (TDSA).

Results have shown that, for problems where the sample space is very large (greater than 500 000 models), e.g. variable selection, TDSA performs well for a large number of simulated and real data sets. Indeed for certain problems it is also possible to apply the optimal proposal methods outlined in the paper. For example, we explored the same autoregressive model choice problem as in the paper, but where we used AIC to distinguish between models. Applying the second-order method it can be seen in this case that the proposal scale changes with the temperature. This gives the nice property that, as the temperature increases, the proposal variance becomes increasingly smaller, adapting to the temperature schedule. The performance of the TDSA algorithm for this problem using the second-order method was again very favourable.

Finally we note that TDSA might be used to tackle many non-statistical problems. In this way reversible jump MCMC methods may permeate a broad range of diverse areas in the same way that MCMC methods and simulated annealing so clearly have.

Peter Green (*University of Bristol*)

I welcome this attempt to provide guidelines for proposal construction in reversible jump Markov chain

Monte Carlo (MCMC) sampling. Being simply the adaptation (I would not even say 'extension') of the Metropolis–Hastings method to variable dimension spaces, it seems safe to expect that this will remain an important approach to 'across-model' MCMC simulation, and, although there are now numerous successful implementations of the idea, clearly researchers need more help in proposal construction. This seems to be curiously difficult to provide.

I wonder whether the authors are tackling the wrong part of the question. The proposal mechanism in equation (5) involves both structural aspects (the choice of the functions $f_{i,j}$ and $v_{i,j,\theta}$) and quantitative ones (the distribution of the random numbers u). In my experience it is the first of these issues that is both more crucial and challenging, whereas the second, that addressed by the authors, is relatively amenable to tuning based on pilot runs.

In understanding the structural aspects of a proposal, it does not necessarily pay to decompose the target into its prior and likelihood terms. Indeed, for many purposes the origin of the variable dimension distribution under study is or should be irrelevant. The situations where it might be relevant are rather special: the rival models need a strong degree of mutual consistency. Suppose that, although different models have quite different parameterizations, there are well-defined functions of parameters with consistent meanings across models—perhaps predictive quantities or 'fitted values'. Further suppose that prior assumptions about such functions are compatible across models. Then these functions are natural candidates for establishing mappings between models that can be used to construct proposals. This, I believe, is the basis for the utility of 'split–merge' and other 'moment matching' methods (for more on the latter, see Green and Richardson (2001)); these work where priors are only weakly informative, and where the matching of moments is sufficient to ensure that likelihoods are close. Incidentally, the split–merge approach is much more flexible (and less myopic) than is commonly realized, as the values of other variables can be freely used in proposals.

Most of the authors' methods are locally formulated, and this is inevitable for analytic methods in all realistic MCMC contexts. Empirical methods, such as the quite naïve but surprisingly effective idea in Section 6 of Green (2002), offer the opportunity of a more global mapping between targets in different models; it could be fruitful to develop both classes further and to make comparisons. I would conjecture that local methods will lose out in multimodal situations.

David Hastie (*University of Bristol*)

This stimulating paper presents several very interesting ideas that will hopefully go some way towards bringing reversible jump Markov chain Monte Carlo algorithms into the domain of the non-expert. I have two small points to contribute to the discussion of this paper.

My first point concerns the zeroth-order method. The higher order methods presented in the paper produce good numerical performance against the 'vanilla' algorithm and are intuitively appealing. However, the same cannot be said for the zeroth-order method which is consistently outperformed in Table 1 and only produces 'more stable estimation of the posterior distribution' (as claimed in Section 8.1) for one out of the two examples presented in Table 2.

Perhaps the authors' observation, that when used with weak non-identifiable centring the zeroth-order method 'may perform poorly when the prior and posterior differ greatly', warns us not to expect too much. None-the-less, it is worth emphasizing that this is exactly when there are sufficient data to tell us something useful about the model.

The second point is again a cautionary one. Using the authors' notation, throughout this paper the function $f_{i,j}$ is assumed to be known. Essentially, this paper introduces methods for optimizing the scaling of a 'local' proposal mechanism $v(u)$ given that a 'global' proposal function $f_{i,j}$ is known. However, consider the following simple example.

Suppose that our target distribution consists of two equally weighted models. Let model 1 be $N(5, 1)$ and model 2 be $N_2\{(5, 5), I_2\}$. Suppose that we choose $f_{1,2}$ and $v(u)$ so that $f_{1,2}\{\theta, v(u)\} = (\theta, Ru)$ where $u \sim N(0, 1)$. Now suppose that we wish to apply the zeroth-order method. Weak non-identifiable centring is not applicable here and so the choice of $b(\theta)$ seems arbitrary. However, choosing $b(\theta) = k \neq 0$ gives $c(\theta) = (\theta, Rk)$ and the resulting equation (8) then becomes difficult to solve for R. The only feasible choice is $b(\theta) = 0$. Then $c(\theta) = (\theta, 0)$ and $R = \exp(25/2)$.

With this proposal mechanism this value of R would give a very poor proposal. The problem occurs because of how we have chosen $f_{1,2}$ and v. To obtain a more sensible proposal we should choose $f_{1,2}\{\theta, v(u)\} = (\theta, 5 + Ru)$. Then, choosing $b(\theta) = 0$, we would obtain $R = 1$ and a good resulting proposal.

Clearly the choice of both f and v is a process that affects the outcome of the methods addressed in this paper. For more difficult examples, it may not be so obvious that we have chosen $f_{i,j}$ or v incorrectly. Do

the authors have any insight into this aspect of proposal design, and have they tried to incorporate any such ideas into the methods presented in this paper?

R. King (*University of Cambridge*)
I congratulate the authors for addressing a difficult issue, within such a general framework. Within Bayesian analyses, where there is model uncertainty, the reversible jump is a powerful tool for simultaneously exploring parameter and model space, although the practical implementation of this procedure can be difficult. The authors present a general methodology which can be applied, possibly removing some of the *ad hoc* reversible jump procedures that are often used. It is also helpful to see the methodology applied to several examples, and the different procedures sensibly compared.

My comments relate to the issue of centring the proposal density within the reversible jump procedure. When using reversible jump Markov chain Monte Carlo sampling, it is my general practice to begin with fairly simple move types and proposal densities. These are then made more complex (if necessary), to improve the mixing of the chain. In my experience, one particularly useful approach that often significantly improves the mixing of the chain is to use an initial pilot Markov chain Monte Carlo run in the 'global' or 'saturated' model, which incorporates all the possible parameters (in the case of the autoregressive example, this would be the AR($k_{\max}$) model), to obtain the posterior means (and possibly variances) of each of the parameters. Then, when proposing to add a parameter in the reversible jump algorithm, the proposal density for the new parameter is centred on the corresponding posterior mean obtained via the pilot run (see for example King and Brooks (2001)). This idea can be extended in some cases, when the global model is not expressed in terms of all possible parameters. For example, see King and Brooks (2002) who apply this approach in the context of capture–recapture data, where models are defined in terms of restrictions placed on the parameters. Of course this procedure assumes that the parameters retain their interpretation for all possible models.

Hans R. Künsch (*Eidgenössiche Technische Hochschule, Zurich*)
My comment concerns the choice of the jump function f_{ij} in the Bayesian model determination problem, i.e. the choice of the manifold where the new proposal will be located (the broken curve in Fig. l(c)). First, I look at the example of the choice of the autoregressive model. Let (X_t) be a stationary Gaussian process with mean 0 and assume that

$$E(X_t|X_{t-1}, \ldots, X_{t-k}) = \sum_{\tau=1}^{k} a_\tau X_{t-\tau},$$
$$\mathrm{var}(X_t|X_{t-1}, \ldots, X_{t-k}) = \sigma_k^2.$$

Then it is easy to see that

$$E(X_t|X_{t-1}, \ldots, X_{t-k-1}) = \sum_{\tau=1}^{k} (a_\tau - \rho_{k+1} a_{k+1-\tau}) X_{t-\tau} + \rho_{k+1} X_{t-k-1}$$

where ρ_{k+1} is the partial correlation between X_t and X_{t-k-1}. In addition, the conditional variance is reduced by the factor $1 - \rho_{k+1}^2$. Hence it seems better to link the two nested models by the function

$$f_{k,k+1}\{(a_\tau), \sigma_k^2, v\} = ((a_\tau - v a_{k+1-\tau}), v, \sigma_k^2(1 - v^2))$$

where v should be restricted to the interval $(-1, 1)$. This proposal distribution will always remain in the subset of parameters corresponding to causal models which I believe is quite an advantage. Explosive autoregressions can usually be excluded *a priori*, and unit roots would require a discrete component in the prior. It is presumably also a better idea to put the prior on the partial autocorrelations than on the autoregressive coefficients directly.

In a more general set-up, the jump function should be chosen such that we jump between distributions which are closest with respect to some distance. From a frequentist point of view, the Kullback–Leibler distance is natural. If the data are produced by model M_j with parameter $\boldsymbol{\theta}_j$ and we fit model M_i by maximum likelihood, we estimate the distribution in M_i which is closest to $L_j(\mathbf{x}|\boldsymbol{\theta}_j)$ in the sense of Kullback–Leibler divergence. In the autoregressive example, by this reasoning we obtain precisely the jump function suggested above, but I do not know whether it is possible to obtain exact or approximate formulae in other, more complex cases.

N. A. Lazar (*Carnegie Mellon University, Pittsburgh*)
I congratulate the authors on a stimulating paper that will no doubt have a significant effect on the way

that reversible jump Markov chain Monte Carlo (MCMC) sampling is used. Although the authors open up many intriguing avenues, I focus my comments on questions relating to model selection.

The marginal likelihood method (Chib, 1995; Chib and Jeliazkov, 2001) offers an alternative to reversible jump MCMC methods. Together with the multitude of techniques for calculating and approximating Bayes factors (DiCiccio *et al.*, 1997), Bayesian statisticians have a variety of tools at their disposal for trans-dimensional model selection. As discussed by DiCiccio *et al.* (1997), computational costs are relevant when evaluating competing methods. Generalizability, as so aptly demonstrated by the present authors, is also important. The proposals given here, especially the lower order approximation methods, when they apply, strike a balance between computation, theory and generalizable implementation.

Assuming that the model dimension or index is also a parameter in the reversible jump MCMC paradigm, I find it somewhat unsettling that the different mechanisms in the paper lead to different posterior model probabilities. Although it is reassuring that the *ordering* stays the same in all the examples reported, if I am interested in the posterior model probabilities in their own right, how should I proceed? How do I know which of the proposals (if any) gives me the 'correct' posterior? In Fig. 4, for example, the 'vanilla' algorithm clearly does not mix well; the auxiliary variable and the correlated auxiliary variable implementations mix better, but they apparently sample the models in different proportions. How should we interpret this? I would appreciate the authors' thoughts on this issue.

Finally, there is model interpretation. The new methodologies allow practitioners to explore a vast array of different, possibly closely related, models. Choosing the 'best' among these, in the sense of highest posterior probabilities, is informative only to the extent that a few models dominate the rest: if the posterior mass is thinly spread across hundreds or thousands of candidates, little is learned. Even in the former case, it might be difficult to give a meaningful interpretation of a specific model chosen from many similar ones. I worry that making reversible jump MCMC sampling too easy might tempt some users to jump into the abyss, without careful thought about which models are reasonable, and what those models mean.

C. Osinski (*Swiss Federal Institute of Technology, Lausanne*)

I report on on-going work that was inspired by this paper. Our aim is to simulate the *a posteriori* distribution for an autoregressive conditional heteroscedastic (ARCH) model where the number of parameters is unknown. We worked on data which were simulated from a given ARCH process. The *a priori* law that we have chosen for such data is the product of the uniform law on $(0, p_{\max})$ for the order of the model, and the log-normal distribution for the parameters of the ARCH(p) process. We decided to allow three different types of jumps: increasing the order, decreasing the order or a classical Markov chain Monte Carlo step within a given order. The move types are chosen according to fixed probabilities depending on the current model, and the proposal law for new parameters is gamma.

Using a reversible jump algorithm without optimizing the proposal law leads to a very poor acceptance rate, and so the ideas of Brooks, Giudici and Roberts become interesting. To apply them, we had to choose a proper way to optimize the acceptance probability, i.e. to choose a central move and an optimizing method as the first-order approximation. In the framework above we cannot do things anyway, or we may find parameters that are inadmissible or we may even try to solve a problem with no solution. Hence the first step is to check that, whatever the data or the *a priori* law are, the problem is well posed. Another difficulty comes when the proposal density has no closed form, for then its parameters must be found iteratively. Using the maximum of the *a posteriori* law as the central move and a first-order approximation, we obtain an acceptance rate that lies between 25% and 50%, which is quite satisfactory, and the convergence does not seem to be problematic. Hence ARCH models illustrate very well how this paper may be useful.

Finally it appears that the paths of the parameters are distributed around the maximum of the likelihood, which is a property that we expect. However, it is known that the maximum likelihood estimators for ARCH models may be quite bad estimators, and so also will be our Bayesian estimates, unless we regularize the likelihood. Hence further investigation is required to obtain an efficient algorithm in this framework, but without the ideas of this paper it seems unrealistic to apply reversible jump Markov chain Monte Carlo methods to ARCH processes.

The **authors** replied later, in writing, as follows.

Firstly, we would like to thank everyone for their contributions to the discussion. Many important points have been raised concerning the choice of f-function, centring function and proposal tuning parameters, as well as broader Markov chain Monte Carlo (MCMC) issues.

Many discussants mention the issue of the choice of an appropriate f-function. This subject is only briefly touched on in our paper, but it is clearly important. Green asks whether we are perhaps addressing

the wrong question in the paper and argues that the choice of the jump function f is more important than the choice of proposal. However, our experience is that pilot tuning reversible jump MCMC samplers is not quite as easy as Green suggests in general, especially where the number of models being considered is large and each jump needs to be individually tuned. Furthermore, sensible adaptive proposal schemes (which allow state-dependent proposal scaling) ought to outperform static proposal parameters chosen on the basis of initial pilot runs. In practice, sensible choices for both f and φ are necessary to construct an efficient chain though, as we shall see later in the context of Hastie's example, the kth-order methods proposed in the paper can compensate for poor choices of jump function, suggesting that, if anything, it is the problem of the choice of proposal that is the most crucial.

McCullagh emphasizes the need to utilize structural relationships between models to guide the construction of f, and we strongly endorse this view. This is most obviously important in the nested model case. However, even then, the choice of f can be a tricky problem, as is highlighted by Forster and Gill. In fact their example represents yet another problem in which posterior correlations (in this case within the saturated model) cause difficulties with MCMC mixing. A simple remedy which would enlighten the construction of an effective f-function involves reparameterization of the saturated model to reduce posterior correlations between components greatly. In their context and others, two reparameterizations would be needed, one fixing X and one fixing Z. This might be undesirable in that a different reparameterization is needed for each move type (i.e. removing X or Z). The projection idea suggested by Forster and Gill therefore seems natural and promising and we were pleased to see that their proposed choice of f together with our second-order method provided such impressive results. We suspect that a similar performance might also be obtained by removing the restriction to deterministic 'down' moves as suggested at the bottom of page 17, so that no parameters remain unchanged when moving from one model to the next. This may also remove the need for the augmentation of model space to include $1+X+Z$, thereby further improving the efficiency.

More generally, as we mention in the paper, and as reiterated by Green, a practical approach to guide the choice of f is to find statistics which have constant interpretation across models, and which can therefore be used to guide the choice of f. The most well-known example of this is the moment matching idea of Richardson and Green (1997) (see also Section 2.1.2), though the general applicability of this technique is still underexplored.

We are particularly interested to hear about the experiences of others with our methods. As well as Forster and Gill, Osinski's example is extremely promising, and it hopefully reinforces our message that the methodology that we describe is easily applied. We hope that these will encourage others to experiment with these techniques though success is of course not guaranteed!

The choice of centring function may or may not be crucial in various applications. Given this, King offers sensible advice for the nested case. Guidance in constructing the centring function on the basis of output from the saturated model is very easy to implement and can be very worthwhile.

Ehlers remarks that in the Gaussian autoregressive example the second-order method effectively samples from the appropriate conditional distribution within the more complex model. One interesting effect of this is that the method is independent of the choice of centring point. This seems to be due to the stability of the second derivative of the log-Gaussian density, and this gives some clues about when the higher order methods are relatively robust to the choice of centring function. In particular, whenever the posterior conditional of the new parameters (conditioning on those that remain fixed) is used as a proposal, the acceptance ratio will be constant and hence all derivatives of the acceptance ratio will be zero whatever the centring function; see Ehlers and Brooks (2002a).

Hastie points out that a badly chosen centring point can lead to poor performance and can mislead the algorithm about the shape of the posterior density. This is only to be expected as properties of the target density at a unique point (the centring point) are being used to approximate the shape of a density function, as noted by Green. For Langevin algorithms in Euclidean spaces these limitations are well known (see for example Roberts and Tweedie (1996)). Practical advice to avoid the worst effect of this kind of problem follows the lines of general MCMC ideas: it is usually sensible to combine more than one type of sampler, at least one of which should be non-adaptive.

It is worth noting that, for Hastie's example, if we take $v = \mu + R\sigma$ and set the first- and second-order log-derivatives of the acceptance ratio to 0 at $b(\theta) = 0$, we obtain $\mu = 5$ and $R = 1$. Thus, our second-order method leads to exactly the same move as recommended by Hastie. In real examples the choice of f may not be quite so obvious (i.e. the value 5 would not generally be known *a priori*), and yet the second-order method would still be able to pick the appropriate location to compensate for Hastie's poor initial choice of jump function. This suggests that the additional complexity (which is trivial in this case) referred to by Robert is perhaps well worth the cost, in general. Of course, when information is available it should

certainly be used to construct the best jump function possible, freeing the kth-order methods to work directly on improving efficiency rather than simply compensating for a poor choice of f.

As Green points out, where possible, it is desirable to use more descriptive global properties of the target density in the design of proposal distributions. In fixed dimensional MCMC sampling, it has proved surprisingly difficult to improve on algorithms which use local information to design appropriate proposals. Of course many examples exist, but none have the generic applicability of Langevin algorithms. In the transdimensional case, many discussants have mentioned promising methodologies for doing this (including Green, Forster and Gill, Dellaportas and Papageorgiou, and Künsch). Although none of these methods will be as practically applicable as our methodology, they provide valuable insights which will stimulate further research.

The choice of the jump function suggested by Künsch in the autoregressive model choice problem is a natural suggestion for the special case where we wish to restrict attention to stationary autoregressive time series. The idea of using a parameterization based on the partial correlations provides a natural way of utilizing the inherent conditional independence structure of the problem. This approach can also be employed in the graphical Gaussian case. In the paper we suggest parameterizing the model in terms of the variance–covariance matrix since our models consist of undirected graphical Gaussian models. However, if we were interested in model determination for directed graphical Gaussian models, we would use a parameterization consisting of regression coefficients, partial variances and partial covariances, as used for instance in Geiger and Heckerman (1994). This approach seems very similar to that suggested by Künsch.

The reparameterization of the autoregressive model in terms of the reciprocal roots, as suggested by Robert, also allows us to focus on stationary processes and he comments on the performance of the zeroth-order method in this case. As with Hastie's example, the naïve use of the zeroth-order method causes problems but the second-order method produces sensible results. For example, if we take a truncated normal (restricted to $[-1, 1]$) proposal for a new real root, the second-order method suggests the posterior conditional (which is also truncated normal) independently of the centring point. Similar results are available for the complex roots (see Ehlers and Brooks (2002b)). A reparameterization from $(\lambda_i, \bar{\lambda}_i)$ to $r\{\cos(\theta) \pm i \sin(\theta)\}$ also provides better results and overcomes Robert's problem of evaluating the acceptance ratio at the centring point with the 'beta-type' proposal. Obvious reparameterizations of this sort are likely to overcome such problems in mosts contexts and, as with any computational method, a little careful thought at the outset will help to avoid many of the potential pitfalls.

The connections between reversible jump methodology in discrete time and the kind of continuous time dynamics described in Møller's contribution are not yet fully understood, but it is clear that more research is needed into synergies between the two areas which have developed almost independently as a result of diverse motivations in Bayesian model choice and the simulation of point processes. To put Møller's contribution in the context of our paper, there is no natural analogue of the zeroth-order algorithm (since there is no accept–reject mechanism) though our kth-order method is motivated by the requirement that the death probability is independent of v and therefore is related to the special case in which the death-rate is chosen to be independent of v. This special case is not necessarily easy to implement in general since it involves needing to identify and separate $b(\theta)$ and $\beta(\theta, v)$ which involves an integral over v. Thus natural analogues of the kth-order procedures for $k \geqslant 1$ exist to obtain the independence of the death-rate as a function of v approximately.

One general point to make concerning the comparison between continuous and discrete time dynamics is the following. The ergodic average of a function f, say, using continuous time dynamics can be written in the form

$$\sum_{i=1}^{\tau} f(X_i)\, a(i) E_i$$

where the $a(i)$s are suitable importance weights and the E_is represent the normalized jump time increments (scaled versions of $\min(T, T')$ in Møller's notation) each with an Exp(l) distribution. Here τ represents the number of jumps achieved by the simulation in the given number of iterations. In general an estimator with a smaller Monte Carlo error is given by

$$\sum_{i=1}^{\tau} f(X_i)\, a(i),$$

that is obtained by just removing the extra (and superfluous) stochasticity of the exponential weighting times. This suggests that a more robust procedure should be obtained by running the jump chain of the continuous time dynamics, and then using appropriate (and deterministic) importance weights to produce the final Monte Carlo estimators.

Meng raises the connections between data augmentation and the use of auxiliary variables. Although both methods are formally identical in the sense that they both involve augmenting the state space of interest to do MCMC sampling, they are motivated very differently (by intractability of the likelihood and by the need to speed up convergence respectively). The main problem with data augmentation methods is that the augmented data are often highly correlated with the parameters of interest. This problem and what to do about it have been extensively investigated by a body of work led by Meng and co-workers (see for example Meng and van Dyk (1999)). In contrast, auxiliary variables are almost always chosen to be independent (or easily transformed to be independent) so difficulties arising from MCMC mixing are not usually caused by correlation between auxiliary variables and parameters.

Lazar comments on the apparent variability of Monte Carlo estimates for posterior model probability and highlights Fig. 4 in particular. In this example, the vanilla algorithm is clearly failing to mix adequately and would require prohibitively large run lengths to provide reliable results (we suspect that Robert may have used different priors from those in Richardson and Green (1997), which may help to explain the difference between his simulation results and our own). The auxiliary variable methods are also subject to Monte Carlo error which can be minimized by taking suitably long runs. We certainly do not believe that either of these two runs is sufficiently long for reliable inference, but they are merely used to demonstrate the dramatic improvement in performance over the corresponding vanilla algorithm. A very simple measure of the Monte Carlo error can be obtained by looking at the variability of the posterior probabilities of interest across replications, as provided in the caption to Table 1 for the autoregressive example. More formal techniques are suggested by Brooks and Giudici (2000) and Brooks *et al.* (2002), though there remains considerable scope for further work in this area.

Some of the contributors mentioned broader MCMC issues. Our focus has very much been on Bayesian model choice. Friel reminds us of the potential for the use of reversible jump methodology in a classical framework as well. Perhaps surprisingly, adaptive MCMC strategies are still not widely used. Andrieu and Doucet describe a promising approach (based on Robbins–Monro optimization techniques) which might provide a framework to allow these methods to become more widely used. The convergence problems described by Erosheva and Feinberg seem largely concerned with within-model mixing, a problem which can often be forgotten somewhat in view of the need to jump between models. In fact transdimensional MCMC methods often alleviate within-model mixing problems by allowing the Markov chain to find a model in which mixing is adequate before returning to a different part of the problematic model space.

As far as model interpretation is concerned (Lazar), posterior model probabilities are best interpreted in the form of posterior odds via Bayes factors if one model is to be selected among a variety of alternatives (Kass and Raftery, 1995). However, if predictive inference is the primary goal then model averaging is most sensible in which case the inference under each model can be combined to provide a single robust estimate irrespectively of the flatness or otherwise of the posterior model space. Certainly providing practitioners with loaded guns, we cannot absolve ourselves of the responsibility for teaching them how to use these tools properly. We agree that careful thought about the range of models under consideration and their meaning in the context of the application are paramount in conducting any statistical analysis, but we think (and perhaps this is even a good thing) that we are still some way from making reversible jump MCMC methods 'too easy' for anyone who is not already fully aware of such issues.

References in the discussion

Al-Awadhi, F., Hurn, M. A. and Jennison, C. (2002) Improving the acceptance rate of reversible jump MCMC proposals. To be published.

Andrieu, C. and Moulines, É. (2002) On the ergodic properties of some adaptive MCMC algorithms. *Technical Report*. University of Bristol, Bristol.

Andrieu, C. and Robert C. P. (2001) Controlled Markov chain Monte Carlo methods for optimal sampling. *Technical Report*. University of Bristol, Bristol.

Andrieu, C. and Robert, C. P. (2002) Controlled MCMC for automatic sampler calibration. *Technical Report*. University of Bristol, Bristol.

Baddeley, A. and Møller, J. (l989) Nearest-neighbour Markov point processes and random sets. *Int. Statist. Rev.*, **2**, 89–121.

Brooks, S. P., Fan, Y. and Rosenthal, J. S. (2002) Perfect forward simulation via simulated tempering. *Research Report*. Cambridge University, Cambridge.

Brooks, S. P., Friel, N. and King, R. (2003) Classical model selection via simulated annealing. *J. R. Statist. Soc.* B, **65**, in the press.

Brooks, S. P. and Giudici, P. (2000) MCMC convergence assessment via two-way ANOVA. *J. Comput. Graph. Statist.*, **9**, 266–285.

Brooks, S. P., Giudici, P. and Philippe, A. (2002) On non-parametric convergence assessment for MCMC model selection. *J. Comput. Graph. Statist.*, to be published.
Cappé, O., Robert, C. P. and Rydén, T. (2002) Reversible jump MCMC converging to birth-and-death MCMC and more general continuous time samplers. Submitted to *J. R. Statist. Soc.* B.
Carlin, B. P. and Chib, S. (1995) Bayesian model choice via Markov chain Monte Carlo methods. *J. R. Statist. Soc.* B, **57**, 473–484.
Chib, S. (1995) Marginal likelihood from the Gibbs output. *J. Am. Statist. Ass.*, **90**, 1313–1321.
Chib, S. and Jeliazkov, I. (2001) Marginal likelihood from the Metropolis–Hastings output. *J. Am. Statist. Ass.*, **96**, 270–281.
Dellaportas, P. (1998) Bayesian classification of Neolithic tools. *Appl. Statist.*, **47**, 279–297.
Dellaportas, P., Forster, J. J. and Ntzoufras, I. (2002) On Bayesian model and variable selection using MCMC. *Statist. Comput.*, **12**, 27–36.
Dellaportas, P. and Papageorgiou, I. (2002) Multivariate mixtures of normals with unknown number of components. To be published.
DiCiccio, T. J., Kass, R. E., Raftery, A. and Wasserman, L. (1997) Computing Bayes factors by combining simulation and asymptotic approximtions. *J. Am. Statist. Ass.*, **92**, 903–915.
van Dyk, D. A. and Meng, X. L. (2001) The art of data augmentation (with discussion). *J. Comput. Graph. Statist.*, **10**, 1–111.
Ehlers, R. S. and Brooks, S. P. (2002a) Efficient construction of reversible jump MCMC proposals for ARMA models. *Technical Report*. University of Cambridge, Cambridge.
Ehlers, R. S. and Brooks, S. P. (2002b) Model uncertainty in integrated ARMA processes. *Technical Report*. University of Cambridge, Cambridge.
Erosheva, E. A. (2002) Grade of membership and latent structure models with application to disability survey data. *PhD Dissertation.* Carnegie Mellon University, Pittsburgh.
Erosheva, E. A., Fienberg, S. E. and Junker, B. W. (2002) Alternative statistical models and representations for large sparse multi-dimensional contingency tables. *Ann. Fac. Sci. Univ. Toul. Math.*, **11**, in the press.
Geiger, D. and Heckerman, D. (1994) Learning Gaussian networks. In *Proc. 10th Conf. Uncertainty in Artificial Intelligence*, pp. 235–243. San Mateo: Morgan Kaufmann.
Godsill, S. (2001) On the relationship between MCMC methods for model uncertainty. *J. Comput. Graph. Statist.*, **10**, 230–248.
Green, P. J. (2002) Trans-dimensional Markov chain Monte Carlo. In *Highly Structured Stochastic Systems* (eds P. J. Green, N. L. Hjort and S. Richardson). Oxford: Oxford University Press.
Green, P. J. and Richardson, S. (2001) Modelling heterogeneity with and without the Dirichlet process. *Scand. J. Statist.*, **28**, 355–375.
Haario, H., Saksman, E. and Tamminen, J. (2001) An adaptive Metropolis algorithm. *Bernoulli*, **7**.
Huerta, G. and West, M. (2000) Bayesian inference on periodicities and component spectral structure in time series. *J. Time Ser. Anal.*, to be published.
Kass, R. E. and Raftery, A. (1995) Bayes factors. *J. Am. Statist. Ass.*, **90**, 773–795.
Kendall, W. S. and Møller, J. (2000) Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Adv. Appl. Probab.*, **32**, 844–865.
King, R. and Brooks, S. P. (2001) On the Bayesian analysis of population size. *Biometrika*, **88**, 317–336.
King, R. and Brooks, S. P. (2002) Bayesian model discrimination for multiple strata capture-recapture data. *Biometrika*, **89**, 785–806.
Manton, K. G., Woodbury, M. A. and Tolley, H. D. (1994) *Statistical Applications using Fuzzy Sets*. New York: Wiley.
Meng, X. L. and van Dyk, D. A. (1999) Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika*, **86**, 301–320.
Meng, X. L. and Schilling, S. (2002) Warp bridge sampling. *J. Comput. Graph. Statist.*, **11**, 485–519.
Møller, J. and Nicholls, G. (1999) Perfect simulation for sample-based inference. *Research Report R-99-2011.* Department of Mathematical Sciences, Aalborg University, Aalborg.
Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. R. Statist. Soc.* B, **59**, 731–792.
Ripley, B. D. (1977) Modelling spatial patterns (with discussion). *J. R. Statist. Soc.* B, **39**, 172–212.
Robbins, H. and Monro, S. (1951) A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400–407.
Roberts, G. O. and Tweedie, R. L. (1996) Exponential convergence of Langevin diffusions and their discrete approximations. *Bernoulli*, **2**, 341–363.
Rosenthal, J. S. (1995) Minorization conditions and convergence rates for Markov chain Monte Carlo. *J. Am. Statist. Ass.*, **90**, 558–566.
Rosenthal, J. S. (2002a) Asymptotic variance and convergence rates of nearly-periodic MCMC algorithms. *J. Am. Statist. Ass.*, to be published.
Rosenthal, J. S. (2002b) Geometric convergence rates for time-sampled Markov chains. *Preprint*. (Available from `http://www.probability.ca/jeff`.)