

# GENERIC REVERSIBLE JUMP MCMC USING GRAPHICAL MODELS

BY DAVID J. LUNN\*, NICKY BEST AND JOHN WHITTAKER

*Imperial College London*

Markov chain Monte Carlo techniques have revolutionized the field of Bayesian statistics. Their enormous power and their generalizability have rendered them the method of choice for statistical inference in many scientific disciplines. Their power is so great that they can even accommodate situations in which the structure of the statistical model itself is uncertain. However, the analysis of such “trans-dimensional” models is not easy, with several significant technical and practical difficulties to overcome. In this paper we present a class of graphical models that allow relatively straightforward analysis of a subset of these trans-dimensional problems. We also present a ‘guided tour’ of the reversible jump methodology underlying our approach and discuss how each of the various difficulties has been circumvented. Our approach has been implemented using the WinBUGS framework as a Gibbs-Metropolis sampling ‘engine’. The main advantage of this is that it affords the analyst much modelling flexibility: trans-dimensional sub-graphs may be used as generic components within an arbitrarily wide range of Bayesian graphical models. We present three example analyses to illustrate our approach.

**1. Introduction.** Markov chain Monte Carlo (MCMC) techniques (Geman and Geman, 1984; Hastings, 1970; Metropolis et al., 1953) have revolutionized the field of Bayesian statistics. Prior to Gelfand and Smith (1990) demonstrating the applicability of MCMC to problems of Bayesian inference in 1990, Bayesian statistics had been a largely academic, and somewhat controversial, pursuit. Since that time, however, a great many applied scientists, in all fields of research, have embraced the ideas behind the Bayesian paradigm. Doubtless many of these subscribe to the underlying philosophy, that all uncertainty is fundamentally subjective, but a substantial proportion have adopted the approach for purely pragmatic reasons. Whatever peoples’ motives, the huge increase over the last decade or so in the number of Bayesian applications appearing in the literature is testa-

---

\*The support of the Medical Research Council (awards G9803841 and G90/82) is gratefully acknowledged.

*AMS 2000 subject classifications:* Primary 62F15, Bayesian inference; secondary 62J12

*Keywords and phrases:* Markov chain Monte Carlo, reversible jump, trans-dimensional model, directed acyclic graph, WinBUGS

ment to the fact that the methodology is now not only generally accepted (although still viewed by some as controversial) but indeed popular. In many fields it has become the method of choice.

Two particular features of MCMC are primarily responsible for this popularity, in our opinion. First, the method is enormously powerful. Gone are the days when our ability to model a system or process of interest is constrained by analytic tractability. MCMC can, *in general*, provide an arbitrarily precise approximation to the exact solution of a given problem, whereas classical approaches tend to focus on delivering an exact solution to an approximation of the problem at hand. With MCMC, arbitrarily complex models can be built and elaborated with ease, without (significantly) affecting one's ability to draw inferences. The MCMC method is so powerful that it can even accommodate situations where the structure of the model itself is the object of inference (Richardson and Green, 1997; Troughton and Godsill, 1998), which is a central theme of this paper.

The other reason for MCMC's rise in popularity is its *generalizability*. The Gibbs sampler, in particular, together with the elegant theory underlying graphical models (Lauritzen et al., 1990; Spiegelhalter, 1998; Spiegelhalter et al., 1993; Whittaker, 1990) provides an algorithm that when viewed abstractly is the same for all models in a very wide class. For every inference problem in this class, the solution can be broken down into a sequence of relatively simple, 'local' computations on the graph. Such simplicity means that it is reasonably straightforward to write one's own software for dealing with specific types of problem. However, this can become laborious: as each new challenge arises, modifications of existing programs are often required, as are entirely new pieces of software from time to time. Also, many researchers lack the time or skills to write their own code, but are specialists in areas where new solutions would be of great scientific value. Hence a need for general purpose software has arisen.

The fact that a Gibbs sampling approach yields the same abstract solution for an entire class of problems might beg the question: "Can we program at an abstract level?" The answer is yes; this is in fact the idea behind *object oriented programming* (Cornell and Horstmann, 1997; Lunn et al., 2000; Reiser and Wirth, 1992). The success of the BUGS software (Lunn et al., 2000; Spiegelhalter et al., 1996b) has been largely due to its exploitation of the conceptual similarities between graphical modelling theory and object oriented programming, which have enabled its authors to produce a very general package that is also 'open-ended' (extensible) – new distributions and functions can be added with relative ease (Lunn, 2003), and specialized interfaces for defining particularly complex models may also be programmed

(Lunn et al., 2002; Thomas et al., 2004). This has made the MCMC method accessible to practitioners in all fields of research, which has widened uptake of the approach. This, in turn, has fuelled evolution of both the methodology and the software, as they have been continually challenged with new model types, and specialists in diverse areas have provided novel insights into practical solutions to existing problems.

One particular type of model still presents significant challenges, however. This is the *trans-dimensional* model, where prior uncertainty regarding the model structure is acknowledged. Thus the model itself becomes a parameter: for example, in “variable selection” we wish to select the ‘best’ subset of all predictor variables on which to regress the response variable – the number of selected variables and the variables themselves, which together define the model structure, are unknown parameters. Such models are of enormous interest in many fields of science as the ability to select the ‘best’ model from a given set and/or to acknowledge uncertainty among those models are key aspects of statistical inference.

The reason why these models are referred to as “trans-dimensional” is that each possible model structure has a potentially distinct set of coefficients/parameters associated with it. In general, these sets will be of different sizes and so as we move from one model to another, the parameter space of interest changes dimension. And here lies the problem with such models. Whilst, *in theory*, the MCMC method can easily accommodate such ‘jumps’ between different parameter spaces, without needing any significant modification, there are several practical issues that can make this difficult in practice. To see this, note that there are two obvious choices as to how to account for the various different models having distinct parameter sets. One approach is to allow our full model to include a separate parameter vector for each possible sub-model (Carlin and Chib, 1995). This leads to difficulties because only the parameter vector associated with the currently selected model has any likelihood. All other parameter vectors are redundant, but they still need to be ‘updated’ (sampled) within the MCMC sampling scheme and, in the absence of a likelihood contribution, their full conditional distributions become equal to their priors, which are typically quite vague. This leads to unrealistic values being sampled for the redundant parameters, which, in turn, can vastly reduce the chances of the relevant models ever being selected subsequently. A preferable approach is to employ a single parameter vector for all models, with length equal to the dimension of the largest parameter space (George and McCulloch, 1996; Green, 1995). In this case the number of *active* components of the parameter vector is variable and the vector can thus be thought of as being of variable dimension.

Each element of the parameter vector plays a potentially different role from iteration to iteration and this has major implications for performing the MCMC simulation. If we have to hand a *reliable* method of proposing new values for the parameters of each model, for example, if their full conditional distributions are available in closed form, then there is no problem. Note, however, that most of the ‘general’ updating methods used in ‘fixed dimension’ MCMC for when the full conditionals are not available in closed form (Gilks, 1992; Hastings, 1970; Metropolis et al., 1953; Neal, 1997) are *adaptive*, that is, they learn from their own (recent) history. If our parameters play different roles from iteration to iteration then their (recent) history is largely irrelevant when considering what their new values should be. Such difficulties in finding ‘general’ methods for updating trans-dimensional models, along with various issues regarding the interpretation of results, which we discuss later, have undoubtedly led to limited software availability. This, in our opinion, inhibits evolution of the methodology.

Of all the approaches to trans-dimensional modelling proposed in recent years, the “reversible jump” method (Green, 1995) appears to be the most effective (Sisson, 2005). This is an example of the single-parameter-vector approach described above and is the method discussed throughout this paper. A principal aim of this paper is to present a class of trans-dimensional models that we have identified as being relatively straightforward to analyse using reversible jump, and to provide a ‘guided tour’ of the methodology. Our intention is to avoid unnecessary mathematical detail but to formally lay down the conditions required for a functioning (valid) algorithm. We also pay particular attention to practical implementation issues that can be easily overlooked but are of great importance when attempting to ‘generalize’ such methods and deliver them to an applied audience. We would also like to dispel the popular misconception that reversible jump is inherently slow – our approach is surprisingly efficient. To maximize the applicability of our approach we formulate the (trans-dimensional) problem of interest as a graphical model. As such it can be slotted into any other graphical structure and the whole may be analysed straightforwardly.

The paper is outlined as follows. In Section 2.1 we describe the class of graphical models of interest and in Section 2.2 we discuss the implied full probability model. In Section 2.3 we present a brief example for clarity, which corresponds to the problem of “variable selection”. In Section 2.4 we discuss prediction from our class of models, which generally amounts to *Bayesian Model Averaging*. Section 3 begins with a ‘guided tour’ of reversible jump MCMC for our class of models, discussing, in turn, the target distribution, the proposal distribution, and the Metropolis acceptance probability. Section

3 concludes with a discussion of how other types of model might be handled. Section 4 discusses practical implementation issues and Sections 5–7 provide three examples of our approach. We conclude with a discussion in Section 8.

## 2. Trans-dimensional models.

2.1. *Graphical models.* The type of graphical model considered in this paper is the *directed acyclic graph* (DAG) (Lauritzen et al., 1990; Whittaker, 1990). Here each quantity in the statistical model is represented by a *node* and arrows are drawn between nodes to show the *direction* of any dependence between the relevant quantities. The graph is *acyclic* if, by following the arrows, it is impossible to return to any given node after leaving it. In what follows we denote the entire set of nodes in the graphical model of interest by  $\mathcal{G}$ . The graphical model itself is denoted by  $\{\mathcal{G}, \mathcal{L}_{\mathcal{G}}\}$  where  $\mathcal{L}_{\mathcal{G}}$  represents the set of directed links. Figure 1 depicts, fairly generally, the type of trans-dimensional sub-model  $\{\mathcal{S}, \mathcal{L}_{\mathcal{S}}\}$  that is considered herein ( $\mathcal{S} \subseteq \mathcal{G}$ ). For notational convenience, we define the set  $\mathcal{R}$  such that  $\{\mathcal{S}, \mathcal{R}\}$  is a partition of  $\mathcal{G}$ .

The notation in Figure 1 is described as follows. Nodes represent both vector and scalar quantities and are denoted by circles. Stochastic and logical (deterministic) dependence between nodes is represented by single- and double-edged arrows respectively. Hence nodes that have double-edged arrows pointing towards them represent functions whereas nodes with single-edged arrows pointing at them are stochastic quantities. A node  $u$  is said to be a *parent* of node  $v$  if an arrow emanating from  $u$  points to  $v$ ; furthermore,  $v$  is then said to be a *child* of  $u$ . However, when attempting to identify probabilistic relationships between the various stochastic nodes in a given graph, all deterministic links are collapsed. For this reason the definitions of parents and children typically correspond to the collapsed graph.

The  $Z$  node in Figure 1 (from whose ancestry we can trace all other nodes in the sub-model) forms the ‘basis’ of the trans-dimensional sub-model: it represents the collection of variables whose joint distribution is in some way dependent on an unknown number of entities, about which we wish to make inferences. The nature of this dependence is either stochastic or logical. In the latter case, the joint distribution of  $Z$  has a fixed number of formal parameters but one or more of those parameters is related deterministically to an unknown number of the entities of interest. For example, one of the most generally applicable types of trans-dimensional model is that used in variable selection, where we wish to choose a subset from a fixed number of covariates on which to regress  $Z$ . Typically we would assume each element

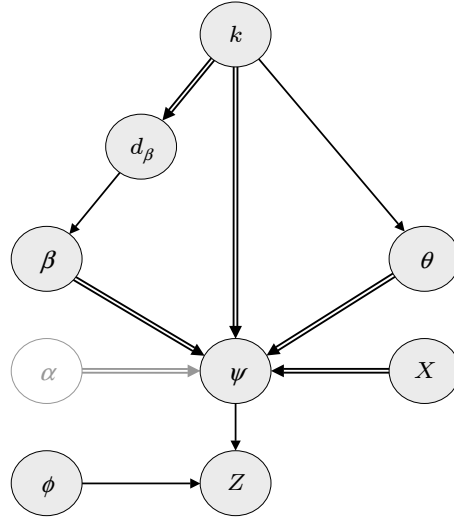


Figure 1: Directed acyclic graph depicting the class of trans-dimensional sub-models  $\{\mathcal{S}, \mathcal{L}_\mathcal{S}\}$  considered herein. Node  $\alpha$  is shown in feint as it represents a trivial extension to the model – see text for details.

of  $Z$  to be a realisation from some distribution (normal, say) whose mean is given by a linear combination of the currently selected variables/covariates.

With stochastic dependence, the joint distribution of  $Z$  has a variable number of formal parameters. For example, in “mixture modelling” we may wish to assume that each element of  $Z$  arises from a (univariate) normal mixture distribution with an unknown number of normal kernels, and hence a variable number of associated means and variances (Richardson and Green, 1997). In such cases, however, we can often introduce auxiliary variables in order to express the model as one involving logical dependence on the entities of interest instead. In mixture modelling, for instance, we can employ auxiliary variables that allocate each element of  $Z$  to a specific kernel. The distribution of each element of  $Z$ , conditional on the allocation variable, is then univariate normal, say, with one mean and one variance, although the kernel to which each  $Z$  is allocated may change from iteration to iteration (of an appropriate MCMC sampler).

The  $\psi$  term in the graphical sub-model of Figure 1 represents *any* logical dependence between  $Z$  and the entities of interest. In cases where stochastic dependence can be re-expressed as logical dependence then any additional auxiliary variables can be simply absorbed, along with their prior parameters, into  $\beta$ , which is described below. For models that cannot be expressed in

this way, the graph in Figure 1 is still applicable if we simply collapse it over  $\psi$  (i.e. delete  $\psi$  and point each of its parents at  $Z$  instead). The  $\phi$  node represents all parents of  $Z$  that have nothing to do with the trans-dimensional part of the model. For example, in the variable selection problem described above, suppose each element of  $Z$  is normally distributed with a common precision parameter  $\tau$ ; then  $\phi = \tau$ . Alternatively, if the joint density of  $Z$  is multivariate normal then  $\phi$  might represent the covariance matrix.

Collectively,  $\beta$ ,  $\theta$ ,  $d_\beta$  and  $k$  are the trans-dimensional core of the sub-model  $\{\mathcal{S}, \mathcal{L}_\mathcal{S}\}$ :  $k$  is the *dimension* of the model, in terms of the number of entities of interest;  $\theta$  is a  $k \times c$  matrix/vector (usually  $c = 1$ ) representing the ‘configuration’ of the model, i.e. its structure – for example, in variable selection, which  $k$  covariates are currently selected;  $d_\beta$  is how many ‘coefficients’ are required for the current dimension/configuration (we only consider models for which this is deterministic – normally  $d_\beta$  is a function of  $k$  only, but we can also allow it to depend on  $\theta$ , in which case a (double-edged) directed link from  $\theta$  to  $d_\beta$  should be added to the graph ( $\theta$  then becomes a stochastic parent of  $\beta$ )); and  $\beta$  is the corresponding  $d_\beta \times 1$  vector of coefficients. Clearly  $\beta$  may be better decomposed into groups of coefficients of different types, such as those with different prior distributions. For example, in the normal mixture model described above, a natural decomposition would be the set of means, the set of variances (or precisions), the set of weights associated with the different kernels, and the (fixed dimension) set of allocation variables (in this case  $\theta$  would be empty). Depicting the model as in Figure 1, however, allows us to discuss these types of model in some generality. In cases where  $\beta$  comprises an intercept term and various gradient parameters associated with a linear predictor, then we may wish to incorporate the intercept into the fixed-dimension part of the model instead, especially as it may require a different prior. This is depicted (in faint) by the introduction of the  $\alpha$  node in Figure 1. However, as this is a fairly trivial extension of the model in which all coefficients are contained in  $\beta$ , we choose to discuss it no further here.

Finally,  $X$  represents the remaining input data that is required in order to fully define the deterministic function  $\psi$ ; for example, in variable selection,  $X$  contains the entire set of covariate data. A particularly attractive feature of our implementation of the reversible jump techniques discussed in this paper is that if elements of  $X$  are unobserved, then arbitrary sub-models for predicting their values can be attached to the graph shown in Figure 1 and we will still be able to perform the required computations. The same also applies to  $Z$  and, where appropriate,  $\phi$ . Thus the sub-model shown in Figure 1 may be inserted into arbitrary other graphs, such as a hierarchical

model, for instance (e.g.  $Z$  may represent a set of unobservable quantities with observed descendants). Note also that arbitrary priors can be specified for  $k$ .

2.2. *Probability model.* The graphical model  $\{\mathcal{G}, \mathcal{L}_{\mathcal{G}}\}$  specifies the joint probability distribution

$$(2.1) \quad p(\mathcal{G}) = p(Z, \phi, \beta, \theta, k, X, \mathcal{R}) = \prod_{v \in \mathcal{G}} p(v | \mathcal{P}_v)$$

(Lauritzen et al., 1990) where the notation  $\mathcal{P}_v$  denotes the set of (stochastic) parents of an arbitrary node  $v$ . Similarly, we use the notation  $\mathcal{C}_v$  throughout to represent the (stochastic) children (or *offspring*) of  $v$ . In what follows it will occasionally be useful to marginalize the joint probability distribution in (2.1) over some of the model parameters, such as  $\beta$  and  $\phi$ . For this reason we introduce some relevant additional notation at this point. Let  $x$  and  $y$  denote arbitrary sets and let  $x \setminus y$  denote ‘all elements of  $x$  except those in  $y$ ’, i.e.  $x \cap y^c$ . Suppose we are interested in integrating the arbitrary quantity  $u$  out of the model to obtain  $p(\mathcal{G} \setminus u)$ . Let the set of nodes  $\mathcal{A}_u$  be defined such that  $\{\mathcal{A}_u, u, \mathcal{C}_u\}$  is a partition of  $\mathcal{G}$ . Then

$$p(\mathcal{G} \setminus u) = \prod_{v \in \mathcal{A}_u} p(v | \mathcal{P}_v) \times \int p(u | \mathcal{P}_u) \prod_{v \in \mathcal{C}_u} p(v | \mathcal{P}_v) du$$

since none of the nodes in  $\mathcal{A}_u$  has a distribution involving  $u$ .

2.3. *An example: variable selection.* To illustrate these types of model further we now describe a version of the variable selection model in detail. It turns out that for a specific choice of prior distribution, we can derive all posterior model probabilities analytically, and so there is no need for any form of MCMC, reversible jump or otherwise, which of course provides us with a convenient way of checking our implementation. Suppose we have  $n$  observations on a response variable of interest  $Z$ , and for each one of these the corresponding values of  $Q$  covariates have been recorded:  $X = \{x_{ij}\}$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, Q$ . Assume that there are no missing data (thus  $X$  is effectively constant) and that the response variable is normally distributed:

$$Z | \tau, \beta, \theta, k \sim \text{MVN}_n(W\beta, \tau^{-1}I_n)$$

where:  $\tau = \phi$  is the residual precision;  $I_n$  denotes the  $n \times n$  identity matrix;  $\beta$  is a  $(k + 1) \times 1$  vector of coefficients (one intercept and  $k$  gradients); and

$W$  is an  $n \times (k + 1)$  design matrix given by

$$W = \begin{pmatrix} 1 & x_{1\theta_1} & \cdots & x_{1\theta_k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n\theta_1} & \cdots & x_{n\theta_k} \end{pmatrix}$$

Hence  $\theta = (\theta_1, \dots, \theta_k)^\top$  is a  $k \times 1$  vector containing the column indices of  $X$  that correspond to the currently selected covariates. Now suppose  $p(\tau, \beta|k) = \text{Gamma}(\tau|a, b) \times \text{MVN}_{k+1}(\beta|\mu, B)$ . If the  $(k + 1) \times (k + 1)$  covariance matrix has the form  $B = \tau^{-1}\Lambda$ , then this represents a *multivariate-normal-gamma* joint prior (Bernardo and Smith, 1994) for  $u = \{\tau, \beta\}$ , which is conjugate with the distribution of the data  $Z$ , and so we can integrate  $\{\tau, \beta\}$  out of the graph analytically.

$$(2.2) \quad p(\mathcal{G} \setminus u) = p(k)p(\theta|k) \int \int p(Z|\tau, \beta, \theta, k)p(\tau, \beta|k)d\beta d\tau$$

The double integral above is the *marginal likelihood*  $p(Z|\theta, k)$ , which is the distribution of the data  $Z$  conditional only on the model structure  $\{\theta, k\}$ . After some algebra one can obtain:

$$(2.3) \quad p(Z|\theta, k) = \frac{\Gamma(a + \frac{n}{2})b^a}{(2\pi)^{n/2}\Gamma(a)} \times |I_n + W\Lambda W^\top|^{-1/2} \\ \times \left[ b + \frac{1}{2}(Z - W\mu)^\top(I_n + W\Lambda W^\top)^{-1}(Z - W\mu) \right]^{-a-n/2}$$

which, incidentally, is a multivariate Student- $t$  density (with  $2a + n - 1$  degrees of freedom) when  $a = b$ .

The joint posterior distribution of  $\theta$  and  $k$ , i.e.  $p(\theta, k|Z)$ , is proportional to (2.2). Thus if the joint state space of  $\theta$  and  $k$ , which is discrete in this case, is not too large we can evaluate the posterior probability of each possible model under any joint prior for  $\theta$  and  $k$  as follows. If there are  $Q$  covariates in total, then there are  $2^Q$  possible models. If we have sufficient computing power we can evaluate (2.2) for each model and subsequently normalize by the sum of all such evaluations to obtain the posterior model probabilities. Bearing in mind that  $2^{10} \approx 1000$ , however, it is clear that this approach would not be possible much beyond  $Q = 30$ , say. Another, arguably more serious, limitation of this approach is that it is only valid when the joint prior for  $\{\tau, \beta\}$  is multivariate-normal-gamma. In our opinion, this is simply not an intuitive prior: aside from the fact that gamma priors for precision parameters can be problematic (Gelman, 2004) and tend to be used purely because of their mathematical convenience, why should we believe

that our prior uncertainty regarding the model coefficients is proportional to the residual variance? This latter concern along with the fact that, more generally, we may wish to allow for missing data and/or incorporate the model into a larger graph requires us to find an alternative way of exploring model space. We achieve this by constructing a Markov chain such that its stationary distribution, marginally, is  $p(\theta, k|\mathcal{D})$ , where, throughout,  $\mathcal{D}$  will denote the observed data, that is the collection of nodes in  $\mathcal{G}$  that form the likelihood ( $\mathcal{D} = Z$  in this example). The reversible jump methodology described in Section 3 facilitates construction of the Markov chain and ensures that it has the correct stationary distribution.

*2.4. Bayesian Model Averaging (BMA).* The very fact that we are considering a model in which the dimension and structure of some sub-model are parameters implies that we acknowledge uncertainty regarding the nature of the underlying ‘true’ model. Except for cases where the nature of the model itself is the only thing of interest, it is against Bayesian philosophy and perhaps somewhat naive to seek to identify some ‘true’ or ‘best’ model from a plausible set, which might then, presumably, be employed to make inferences about other quantities of interest. Any such inferences should, ideally, fully reflect the extent of (posterior) model uncertainty. Following on from the previous example:

$$(2.4) \quad p(\Delta|\mathcal{D}) = \sum_{\{\theta, k\} \in \mathcal{M}} p(\Delta, \theta, k|\mathcal{D}) = \sum_{\{\theta, k\} \in \mathcal{M}} p(\Delta|\theta, k, \mathcal{D})p(\theta, k|\mathcal{D})$$

where  $\Delta$  is some *predictive* quantity of interest, some future observations, say, or the loss/utility associated with some decision, and  $\mathcal{M}$  denotes the discrete state space of  $\{\theta, k\}$ . (More formally, we define a *predictive* quantity as any quantity that is not an essential part of the graphical model, that is, it has no descendants among  $\mathcal{D}$ . The posterior distributions of all essential (non-predictive) quantities are evaluated as a matter of course during the analysis anyway.) The above is referred to as *Bayesian Model Averaging* (BMA) (Raftery et al., 1997): Equation (2.4) is a weighted average of the posterior distributions for the quantity of interest under each possible model, with each posterior weighted by the posterior model probability  $p(\theta, k|\mathcal{D})$ . In the case of variable selection, we have discussed the calculation of each  $p(\theta, k|\mathcal{D})$  in the previous subsection; if  $\Delta$  represents a set of future observations then  $p(\Delta|\theta, k, \mathcal{D}) = p(\Delta|\theta, k)$  has the form of (2.3) with  $Z = \Delta$ . As the right-hand-side of (2.4) is a function of  $p(\theta, k|\mathcal{D})$ , its evaluation, in general, will suffer from (at least) the same practical problems as those encountered in evaluating  $p(\theta, k|\mathcal{D})$  – the state space of  $\{\theta, k\}$  is potentially

enormous and the integration (with respect to  $\beta$  and  $\tau$ ) required to obtain the marginal likelihood  $p(\mathcal{D}|\theta, k)$  is generally only straightforward if  $\beta$  and  $\tau$  have a conjugate joint prior. However, if we can sample from the posterior distribution then we can evaluate  $p(\Delta|\mathcal{D})$  by Monte Carlo integration. From (2.4):

$$(2.5) \quad P(\Delta|\mathcal{D}) = \mathbb{E}_{p(\theta, k|\mathcal{D})} [p(\Delta|\theta, k, \mathcal{D})] \approx \frac{1}{T} \sum_{t=1}^T p(\Delta|\theta^{(t)}, k^{(t)}, \mathcal{D})$$

where  $\{\theta^{(t)}, k^{(t)}, t = 1, \dots, T\}$  is a sample of size  $T$  from  $p(\theta, k|\mathcal{D})$ . Hence we can approximate  $p(\Delta|\mathcal{D})$  ‘on the fly’ during an MCMC run by simply drawing from  $p(\Delta|\theta, k, \mathcal{D})$  for each posterior sample of  $\{\theta, k\}$ .

More generally,  $\theta$  may have a continuous state space, in which case (2.5) still holds. Also,  $\Delta$  may be most naturally expressed in terms of quantities other than, or additional to,  $\theta$  and  $k$ . For example, prediction based on  $\psi' = \psi(\beta, \theta, k, X')$ , where  $X'$  denotes (pre-specified) future input data (such as new covariate measurements), is a common goal in applying these techniques. If we denote the parents of  $\Delta$ , which may include elements of  $\mathcal{D}$ , by  $\mathcal{P}_\Delta$  then we can re-write Equation (2.5), in its most useful form, for arbitrary quantities of interest  $\Delta$ :

$$(2.6) \quad p(\Delta|\mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^T p(\Delta|\mathcal{P}_\Delta^{(t)})$$

where the dependence on  $\mathcal{D}$  has been lost through the fact that if  $\Delta$  is ‘predictive’ and thus has no descendants among  $\mathcal{D}$ , then it is conditionally independent, given its parents, of all elements of  $\mathcal{D}$  that are not among its parents (Lauritzen et al., 1990). The significance of (2.6) is that it demonstrates that the posterior distribution of any predictive quantity can be approximated and/or sampled from trivially given a set of posterior samples for its parents. Hence, fully Bayesian inference on  $\Delta$ , acknowledging uncertainty regarding the structure (and dimension) of the model, can be achieved by simply incorporating nodes into the graphical model that represent both  $\Delta$  and any ancestors of  $\Delta$  not already present, and by then *forward sampling* those nodes, i.e. sampling each conditional on its parents in any order such that a node is only sampled when all of its ancestors have been sampled (i.e. down the graph).

There is a serious practical issue here: we have not really circumvented the problem that the number of possible models may be vast. Although we can construct a Markov chain that will move around model space according to  $p(\theta, k|\mathcal{D})$ , the number of possible models may be such that we cannot

hope to even attempt to visit every state. Hence, it is feasible that we may miss some, or even all, of the best models. However, it is often reasonable to assume that the majority of probability mass is localized to specific areas of model space and that the distribution is reasonably smooth in those areas, such that there are no isolated ‘spikes’ of probability. Our inferences should be reasonably accurate so long as we can find and explore these localizations. We aim to achieve this by ensuring our sampler can make both local and middle/long distance moves.

### 3. Reversible Jump MCMC.

3.1. *The target distribution.* Our approach throughout is based on a Gibbs sampling scheme. All nodes associated with the ‘fixed dimension’ part of the model are updated by standard means; for example, specialized random number generators for closed-form full conditionals, and standard Metropolis and slice sampling techniques (among others) for more general distributions (Lunn et al., 2000). This part of the updating loop is not of interest here, however. In this paper, we are concerned only with the ‘variable dimension’ part of the graph, that is  $k$ ,  $\theta$  and  $\beta$ . Hence we seek a strategy for drawing from  $p(\beta, \theta, k | \mathcal{R}, X, \phi, Z) \propto p(\mathcal{G})$ . Note that we can re-express  $p(\mathcal{G})$  as follows:

$$\begin{aligned}
 p(Z, \phi, \beta, \theta, k, X, \mathcal{R}) &= p(\beta | Z, \phi, \theta, k, X, \mathcal{R}) \times p(Z, \phi, \theta, k, X, \mathcal{R}) \\
 &= \text{FC}(\beta) \times p(\mathcal{G} \setminus \beta) \\
 &= \text{FC}(\beta) \times \prod_{v \in \mathcal{A}_\beta} p(v | \mathcal{P}_v) \times \int p(\beta | k) p(Z | \phi, \beta, \theta, k, X) d\beta \\
 &= \text{FC}(\beta) \times p(Z | \phi, \theta, k, X) \times \prod_{v \in \mathcal{A}_\beta} p(v | \mathcal{P}_v)
 \end{aligned}$$

where  $\text{FC}(x)$  denotes the full conditional distribution of  $x$ . Here we are assuming that  $\beta$  has no ‘external’ (i.e. outside  $\{\mathcal{S}, \mathcal{L}_\mathcal{S}\}$ ) children:  $\mathcal{C}_\beta \cap \mathcal{R} = \emptyset \Rightarrow \mathcal{C}_\beta = Z$ . This is quite reasonable as for  $\beta$  to have any further offspring would require it to also form part of another trans-dimensional model, which is difficult to imagine (note that any predictive quantities that may be dependent on  $\beta$ , such as  $\Delta$  in the previous sub-section, do not count, *technically*, as children of  $\beta$ , since they are not essential for the model’s analysis and so are not part of the graph proper). Our target distribution, up to a constant of proportionality, is thus given by

$$\begin{aligned}
 (3.1) \quad \text{FC}(\beta, \theta, k) &= p(\beta, \theta, k | Z, \phi, X, \mathcal{R}) \\
 &\propto \text{FC}(\beta) \times p(Z | \phi, \theta, k, X) \times \prod_{v \in \mathcal{H}} p(v | \mathcal{P}_v)
 \end{aligned}$$

where  $\mathcal{H}$  represents the set of nodes whose distributions involve  $\theta$  or  $k$  (or both) but not  $\beta$ . More formally,  $\mathcal{H} = \{\theta, k\} \cup \{(\mathcal{C}_\theta \cup \mathcal{C}_k) \cap \mathcal{R}\}$ , that is,  $\theta$ ,  $k$ , and all external children of  $\theta$  and  $k$ . Typically,  $\theta$  and  $k$  have no children outside  $\{\mathcal{S}, \mathcal{L}_\mathcal{S}\}$  and so  $\mathcal{H}$  is simply  $\{\theta, k\}$ . In what follows we will assume this to be the case, unless we state otherwise. We discuss the significance of the factorization in (3.1) later, when we examine the Metropolis-Hastings acceptance probability in Section 3.3.

### 3.2. The proposal distribution.

3.2.1. *Notation.* At this stage it is convenient to define a mapping from  $\{\beta, \theta, k\}$  to a set  $\{\zeta, M\}$  such that  $M$  represents a discrete, scalar, model identifier and  $\zeta$  represents a vector of continuous model parameters that has fixed dimension conditional on the value of  $M$ ,  $d_m = \dim(\zeta|M = m)$ . More formally,

$$\{\zeta, M\} = \{f_\zeta(\beta, \theta), f_M(\theta, k)\}$$

where the nature of  $f_\zeta(\cdot, \cdot)$  and  $f_M(\cdot, \cdot)$  depend on whether  $\theta$  is discrete or continuous. Typically,  $\theta$  is discrete and the structure of the trans-dimensional model is uniquely identified by  $\theta$  and  $k$  together. Hence  $M = f_M(\theta, k)$  is a one-to-one mapping, which is therefore invertible, such that  $\{\theta, k\} = f_M^{-1}(M)$ . In this case  $\zeta$  is simply given by  $f_\zeta(\beta, \theta) = \beta$ . If, on the other hand,  $\theta$  is continuous (see Section 6, for example), then the structure of the trans-dimensional model is identified by  $k$  alone, and  $\theta$  becomes part of  $\zeta$  as it now represents a collection of model parameters. Hence

$$\zeta = \{\beta, \theta\}; \quad M = k$$

In either case ( $\theta$  continuous or discrete), the distribution of  $\{\zeta, M\}$  is equivalent to that of  $\{\beta, \theta, k\}$  and so we can concentrate on proposing values for  $\{\zeta, M\}$  instead. This is convenient since

$$\pi(\zeta, M = m) = \pi(\zeta|M = m) \times \Pr\{M = m\}$$

(where  $\pi$  is some arbitrary probability measure of interest). Hence, the distribution of our complex, ‘variable dimension’ quantity of interest can be factorized into the product of a single probability and a conditional probability density defined on  $\mathbb{R}^{d_m}$ , which is a space of fixed dimension. We exploit this fact when constructing our proposal distribution for  $\{\zeta, M\}$  – we first propose a new model identifier,  $M = m'$ , say, and then a new set of model parameters  $\zeta'$  from  $\mathbb{R}^{d_{m'}}$  (or some appropriate subspace thereof).

3.2.2. *Requirements of Metropolis-Hastings.* The main difference (indeed the only *significant* difference) between reversible jump and standard Metropolis-Hastings (M-H) approaches (Hastings, 1970; Metropolis et al., 1953) is that we entertain the idea of moves between probability spaces of different dimension. Indeed, in order to derive the acceptance probability for the types of model that we consider herein, we follow largely the same steps as in Hastings (1970) – careful choice of notation is required, however, as is a substantial degree of mathematical rigour (see Waagepetersen and Sorensen (2001) for a reasonably accessible derivation). We do not wish to derive the acceptance probability in this paper; instead, to aid intuition, we merely provide some discussion of the key requirements, as follows. The “jump” part of the “reversible jump” name stems from the fact that we attempt to jump between spaces of different dimensions. The “reversible” part of the name has a somewhat less obvious origin. The probability distribution of the Markov chain that we construct must, after convergence, be *invariant* (or *stationary*). A sufficient (but not necessary) condition for invariance is *reversibility*. This means that the joint distribution of each pair of consecutive states is the same as it would be if the chain were ‘time-reversed’. In other words, for any pair of consecutive states (after convergence) the probability of being in the current state and then moving to the next state is the same as that of first being in the next state and then moving to the current state. The same sufficient condition (reversibility) is used to derive the acceptance probability for standard M-H, and so reversibility does not really distinguish reversible jump from its ‘fixed dimension’ counterpart. However, in standard M-H the nature of the proposal distribution typically ensures reversibility automatically. By contrast, in reversible jump we must define our proposal distribution in terms of complementary ‘jumps’, such as “birth” and “death”. Whenever we allow a change of dimension, we must ensure that our proposal distribution permits the same dimension change in reverse. We must also ensure that we can return to the original set of parameter values. This latter concern is dealt with by always defining new parameter values in terms of the original parameter values and a set of random variables, via a one-to-one mapping. As all one-to-one mappings are invertible, the reverse move is always well defined.

A necessary condition for the existence of a one-to-one mapping between the quantities of interest is the following ‘dimension matching condition’ (Green, 1995). Consider the (proposed) move  $\{\zeta^{(t)}, m^{(t)}\} \rightarrow \{\zeta', m'\}$ , where  $\zeta^{(t)}$  and  $m^{(t)}$  denote the current values (at iteration  $t$ ) of  $\zeta$  and  $M$ , respectively. The dimension matching condition is

$$\dim(\zeta^{(t)}) + \dim(r^{(t)}) = \dim(\zeta') + \dim(r')$$

where  $r^{(t)}$  and  $r'$  are the vectors of random variables required in order to map  $\zeta^{(t)}$  to  $\zeta'$  and vice versa, respectively (note that  $\dim(M) = 1$  always). Thus the dimension of the current model plus the number of random variables required to map to an appropriate set of parameters for the new model must be preserved between each possible jump and the reverse move. The mapping itself is denoted as follows:

$$\{\zeta', r'\} = g_{m^{(t)}m'}(\zeta^{(t)}, r^{(t)}) \Rightarrow \{\zeta^{(t)}, r^{(t)}\} = g_{m^{(t)}m'}^{-1}(\zeta', r') \\ [= g_{m'm^{(t)}}(\zeta', r')]$$

This may look complicated but in many situations we wish to propose new values for  $\zeta$  directly, that is, independently of the current state ( $\zeta^{(t)}$ ). In such cases — for example, when the full conditional distribution for  $\zeta$  is available in closed form — we simply set  $\zeta' = r^{(t)}$  and  $r' = \zeta^{(t)}$ ; thus  $g_{m^{(t)}m'}(\cdot, \cdot)$  represents an ‘identity mapping’:

$$\begin{bmatrix} \zeta' \\ r' \end{bmatrix} = I_{(d_{m^{(t)}}+d_{m'})} \times \begin{bmatrix} r^{(t)} \\ \zeta^{(t)} \end{bmatrix}$$

3.2.3. *The proposal density.* In general, we write the proposal distribution for the move  $\{\zeta^{(t)}, m^{(t)}\} \rightarrow \{\zeta', m'\}$  as

$$\text{Prop}(\{\zeta^{(t)}, m^{(t)}\} \rightarrow \{\zeta', m'\}) = q(m^{(t)} \rightarrow m') \times q_{m^{(t)}m'}(r^{(t)}, \zeta^{(t)})$$

where, throughout,  $q(\cdot)$  denotes the probability according to our proposal scheme of the indicated *discrete* move. Hence  $q(m^{(t)} \rightarrow m')$  is the probability of deciding to attempt a move from model  $m^{(t)}$  to model  $m'$ . The term  $q_{m^{(t)}m'}(r^{(t)}, \zeta^{(t)})$  then denotes the conditional proposal density for  $r^{(t)}$ , which is defined on  $\mathbb{R}^{\dim(r^{(t)})}$  (the dependence on  $\zeta^{(t)}$  is due to the fact that any M-H proposal is allowed to depend on the current state). In this paper, we mainly consider situations in which the full conditional distribution for  $\beta$  is available in closed form; we then use this full conditional for proposing new vectors  $\beta'$  conditional on  $k = k'$  and  $\theta = \theta'$  (note that this necessitates a sampling order of  $k, \theta, \beta$ , which is natural given the structure of the graphical model shown in Figure 1). Thus we write  $r^{(t)} = \{\beta', r_\theta^{(t)}\}$  and  $q_{m^{(t)}m'}(r^{(t)}, \zeta^{(t)})$  becomes

$$p(\beta' | Z^{(t)}, \phi^{(t)}, \theta', k', X^{(t)}, \mathcal{R}^{(t)}) \times q_{m^{(t)}m'}(r_\theta^{(t)}, \zeta^{(t)}),$$

where  $r_\theta^{(t)}$  is empty (and  $q_{m^{(t)}m'}(r_\theta^{(t)}, \zeta^{(t)}) \equiv 1$ ) unless  $\theta$  is continuous, in which case  $r_\theta^{(t)}$  represents the vector of random variables required to map

$\theta^{(t)}$  to  $\theta'$ , given  $M = m'$ . The proposal density for the reverse move, in full, is given by

$$\begin{aligned} \text{Prop}(\{\zeta', m'\} \rightarrow \{\zeta^{(t)}, m^{(t)}\}) &= q(m' \rightarrow m^{(t)}) \\ &\times p(\beta^{(t)} | Z^{(t)}, \phi^{(t)}, \theta^{(t)}, k^{(t)}, X^{(t)}, \mathcal{R}^{(t)}) \\ &\times q_{m'm^{(t)}}(r'_\theta, \zeta') \end{aligned}$$

Our reason for mainly restricting attention to (partially) conjugate models is, basically, that we must have a *good* proposal distribution for the coefficients in order to stand any reasonable chance of accepting the new state. At first glance, it looks as though we might try using some form of multivariate normal kernel, say, as a proposal distribution for  $\beta$ , since we are employing an M-H algorithm anyway. However, such an approach would be fraught with problems. The important thing to note is that it is the very nature of trans-dimensional models that the vector of coefficients  $\beta$  can have a different meaning/interpretation for each possible value of  $M$ . Moreover, coefficients associated with each entity of interest can, and do, move around the  $\beta$  vector as the entities of interest are included in and discarded from the model configuration. Unless the proposed model configuration is the same as, or is a reduced version of, the current configuration, there is no well-defined ‘current point’ around which to construct a reliable kernel, and without some complex and memory-intensive housekeeping there is no possibility of allowing the kernel to adapt itself based on previous behaviour of the Markov chain. As we will see shortly, making use of the full conditional for  $\beta$  allows us to write the M-H acceptance probability in terms of  $\theta$  and  $k$  only, which means that the effectiveness of our approach depends only on our strategy for exploring  $\{\theta, k\}$ -space. Trans-dimensional models for which the full conditional distribution of  $\beta$  is not available in closed form are discussed in Section 3.4.

Under the above strategy the dimension matching condition is automatically satisfied for discrete  $\theta$ ; in the case of continuous  $\theta$  it is reduced to

$$\begin{aligned} \dim(\theta^{(t)}) + \dim(r_\theta^{(t)}) &= \dim(\theta') + \dim(r'_\theta) \\ (3.2) \quad \Rightarrow k^{(t)}c + \dim(r_\theta^{(t)}) &= k'c + \dim(r'_\theta) \end{aligned}$$

The choice of  $q_{m_1 m_2}(r_\theta, \zeta)$  for cases where  $\theta$  is continuous is context-dependent. In Section 6,  $\theta$  represents a set of continuous ‘knot-points’ in a ‘piecewise polynomial’ regression ( $\dim(\theta) = k$ , i.e.  $c = 1$ ). In this case, when proposing a new knot,  $r_\theta^{(t)}$  contains two random variables, one specifying which of the  $k' = k^{(t)} + 1$  possible positions in  $\theta'$  the new knot is to occupy,

and another specifying the new knot's value. Both of these are chosen uniformly from the full range of possibilities available. For the reverse move,  $r'_\theta$  is a scalar and simply represents the position in  $\theta'$  to be deleted. Except for  $q_{m_1 m_2}(r_\theta, \zeta)$ , all that remains to be specified for a fully defined proposal distribution is the method for choosing moves between models, i.e.  $m^{(t)} \rightarrow m'$ , which we discuss below.

**3.2.4. Move types.** In the current implementation of our approach, in WinBUGS (Lunn et al., 2000; Spiegelhalter et al., 1996b), there are four possible move types: ‘birth’; ‘death’; ‘replace’; and ‘do nothing’. The first step in our proposal is to choose one of these at random, subject to appropriate constraints that keep  $k$  within the minimum and maximum permitted values. These are denoted by  $k_{\min}$  ( $\geq 0$ ) and  $k_{\max}$  ( $< \infty$ ), respectively, and are defined either by the prior or by the nature of the problem at hand). In ‘birth’ and ‘death’ we first generate a positive integer  $\delta$  from some appropriate distribution and use it to increase or decrease  $k$ ; again, we wish to ensure that  $k_{\min} \leq k \leq k_{\max}$  always, and so we constrain the distribution of  $\delta$  accordingly. We set  $k' = k^{(t)} + \delta$  and  $k' = k^{(t)} - \delta$  for ‘birth’ and ‘death’ respectively. This completes the specification of  $m^{(t)} \rightarrow m'$  for continuous  $\theta$ , but we must still modify  $\theta$  conditional on the new value of  $k$ , as we must in the case of discrete  $\theta$  to fully define  $m^{(t)} \rightarrow m'$ . In ‘death’ we typically choose  $\delta$  of the existing  $k^{(t)}$  elements of  $\theta^{(t)}$  to be deleted: we usually do this uniformly so that the associated (proposal) probability is  $\delta!(k^{(t)} - \delta)!/k^{(t)}!$ . In ‘birth’ we choose  $\delta$  new elements for  $\theta$  from the available possibilities. When  $\theta$  is continuous, the dimension matching condition (3.2) dictates that we must generate  $2\delta$  random numbers in the process since we generate  $\delta$  in the reverse move (‘death’) – one possibility is to generate both a value and a position in  $\theta'$  for each new element, as discussed above for the spline-fitting example of Section 6 (note that the positions generated then directly define the reverse ‘death’ move). Note that ‘birth’ and ‘death’ here are abstract concepts used to represent the more general processes of increasing and decreasing  $k$ , respectively. For some applications it may be desirable to extend ‘birth’ and ‘death’ to include other types of move, such as ‘split’ and ‘combine’, which are useful in mixture modelling (Richardson and Green, 1997), for example.

In ‘replace’ moves we typically delete a number of elements of  $\theta^{(t)}$  (as in a ‘death’ move) and then introduce the same number of new elements (as in ‘birth’). We first choose the number  $\delta$  of elements to replace (applying all appropriate constraints, as usual) and then we perform the necessary ‘birth’ and ‘death’ moves in such a way that the new elements are distinct from the

deleted ones. There is no real need to impose this latter constraint, it merely eliminates the possibility of redundant (or partially redundant) moves.

The fact that  $\delta$  is allowed to be greater than one means that our sampler is capable of making both local and middle/long distance moves in  $\{\theta, k\}$ -space. This, we believe, improves our chances of fully exploring the posterior distribution, since there will likely be more than one localization of posterior probability within  $\{\theta, k\}$ -space and allowing  $\delta$  to exceed one may enable us to jump from one such localization to another.

The idea behind the ‘do nothing’ move is that it provides an opportunity to update the elements of  $\beta$  via standard Gibbs steps without changing the structure of the model. This is particularly useful for situations in which only selected elements of  $\beta$  tend to be updated whenever the model structure is modified, via a ‘birth’, ‘death’ or ‘replace’ move; for example, when the full conditional for  $\beta$  is not available in closed form (see Section 3.4). Such ‘within-model’ sampling is essential for adequate mixing of the Markov chain.

3.3. *The acceptance probability.* The M-H acceptance probability for the proposal strategy defined above is given by  $\rho = \min\{1, \rho' J\}$ , where  $J$  is a Jacobian term arising from the change of variables  $\{\zeta', r'\} = g_{m^{(t)}m'}(\zeta^{(t)}, r^{(t)})$  and is given by

$$J = \left| \det \left\{ \frac{\partial g_{m^{(t)}m'}(\zeta^{(t)}, r^{(t)})}{\partial (\zeta^{(t)}, r^{(t)})} \right\} \right|,$$

and

$$\rho' = \frac{p(\beta', \theta', k' | Z^{(t)}, \phi^{(t)}, X^{(t)}, \mathcal{R}^{(t)})}{p(\beta^{(t)}, \theta^{(t)}, k^{(t)} | Z^{(t)}, \phi^{(t)}, X^{(t)}, \mathcal{R}^{(t)})} \times \frac{\text{Prop}(\{\zeta', m'\} \rightarrow \{\zeta^{(t)}, m^{(t)}\})}{\text{Prop}(\{\zeta^{(t)}, m^{(t)}\} \rightarrow \{\zeta', m'\})}$$

For all of the models analysed in this paper the function  $g_{m^{(t)}m'}(\cdot, \cdot)$  represents an identity mapping and so  $J = 1$ . Using the factorization given in (3.1) along with the expressions for the proposal density given in Section 3.2.3, we can rewrite  $\rho'$  as

$$\begin{aligned} & \frac{p(Z^{(t)} | \phi^{(t)}, \theta', k', X^{(t)}) \times p(k' | \mathcal{P}_k^{(t)}) \times p(\theta' | k', \{\mathcal{P}_\theta \setminus k\}^{(t)})}{p(Z^{(t)} | \phi^{(t)}, \theta^{(t)}, k^{(t)}, X^{(t)}) \times p(k^{(t)} | \mathcal{P}_k^{(t)}) \times p(\theta^{(t)} | \mathcal{P}_\theta^{(t)})} \\ & \times \frac{q(m' \rightarrow m^{(t)}) \times q_{m'm^{(t)}}(r'_\theta, \zeta')}{q(m^{(t)} \rightarrow m') \times q_{m^{(t)}m'}(r_\theta^{(t)}, \zeta^{(t)})} \end{aligned}$$

Note that the factors corresponding to the  $\beta$ -full-conditionals in each target and proposal distribution pair cancel each other out. Also note that usually  $\theta$  is discrete and so the  $q_{m_1 m_2}(r_\theta, \zeta)$  terms disappear. Further, if  $\theta$  is

continuous, although  $q_{m_1 m_2}(r_\theta, \zeta)$  is technically allowed to depend on  $\zeta$ , in the vast majority of situations dependence on  $\theta$  alone will suffice; that is, conditional on  $M = m'$ , the proposal density for  $\theta^{(t)} \rightarrow \theta'$  depends only on  $\theta^{(t)}$ . Hence the acceptance probability is almost always independent of  $\beta$  (both  $\beta^{(t)}$  and  $\beta'$ ), which has two useful consequences. First, it means that we can save time by only proposing  $\beta'$  if we have already decided to accept the new state. Second, and more importantly, it means that our decision to move from one state to another is based solely on the corresponding model configurations – the parameters themselves are irrelevant; thus the effectiveness of our approach depends only on our strategy for exploring the joint probability space of  $\{\theta, k\}$ , which we believe is quite reasonable.

The reasons why our acceptance probability is independent of  $\beta$  are as follows. First, as mentioned above, our proposal distribution for  $\beta$  is the full conditional, which cancels with the full conditional in the expression for the target distribution given by (3.1). Second, any remaining  $\beta$ -dependence has been integrated out of the target distribution via  $p(Z|\phi, \theta, k, X) = \int p(\beta|k)p(Z|\phi, \beta, \theta, k, X)d\beta$ . A straightforward way to calculate this integral is to first note that the integrand is proportional to the full conditional distribution of  $\beta$ , which can be expressed in closed form. Identifying this closed form by standard means (Gilks, 1996) will also reveal the constant of proportionality, which is all that will be left after integrating over the state space of  $\beta$ . Thus the significance of the factorization in (3.1) is now revealed: the first term disappears due to cancellation; the second term can be calculated as a by-product of deriving the full conditional distribution for  $\beta$ , which we require anyway; and the third term is simply the product of priors for  $\theta$  and  $k$ . To illustrate the calculation of  $p(Z|\phi, \theta, k, X)$  we return to the variable selection problem described in Section 2.3. The full conditional distribution of  $\beta$  is  $\text{MVN}_{k+1}(\chi, V)$ , where  $V^{-1} = B^{-1} + \tau W^T W$  and  $\chi = \tau V W^T Z$ . The integrand of interest is therefore  $A \times \text{MVN}_{k+1}(\beta|\chi, V)$  for some ‘constant’  $A$ , which integrates to  $A$ . Hence

$$\begin{aligned} p(Z|\phi, \theta, k, X) &= A = \frac{p(\beta|k)p(Z|\phi, \beta, \theta, k, X)}{\text{MVN}_{k+1}(\beta|\chi, V)} \\ &= \left( \frac{\tau^n |V|}{2^n \pi^n |B|} \right)^{1/2} \times \exp \left[ -\frac{1}{2} \left( \tau Z^T Z - \chi^T V^{-1} \chi \right) \right] \end{aligned}$$

It turns out that we can, in general, exploit the machinery that already exists within WinBUGS for deriving (closed form) full conditional parameters (Lunn et al., 2000) in order to evaluate  $p(Z|\phi, \theta, k, X)$  in this way.

**3.4. Model types.** Types of trans-dimensional model for which the full conditional distribution of  $\beta$  may be available in closed form include splines,

variable selection, ‘unknown order’ Markov chains, and certain types of partition model, for example. This class is widened substantially by the availability of various auxiliary variable techniques. As we shall see in Section 6, the introduction of normally distributed auxiliary variables into problems involving binary data can greatly facilitate probit regression. The same approach can be adapted for handling ordered categorical data. Logistic regressions (on binary data) are made possible by the further introduction of Kolmogorov-Smirnov random variables (Holmes and Held, 2005) and this approach has a natural extension to problems involving *unordered* categorical data. Exponentially distributed auxiliary variables can be used to similar effect in Poisson regression (Frühwirth-Schnatter and Wagner, 2004).

While the above class of models is reasonably sized, we have no intention of being tied to conjugate models alone. The approach described in Sections 3.1–3.3 relies somewhat on the availability of  $FC(\beta)$  in closed form, but it is quite straightforward to evaluate the Metropolis acceptance probability for *any* modelling scenario. As mentioned earlier, the difficulty with reversible jump lies in constructing an ‘instantly reliable’ proposal distribution for the model coefficients. While this may be more difficult for non-conjugate models, it is by no means impossible – see Richardson and Green (1997) and Brooks et al. (2003), for example. Conditional on the availability of a reliable proposal distribution, the reversible jump algorithm itself is simple, and we have exploited this fact in implementing our approach within WinBUGS. Arbitrary model types and proposal distributions are permissible (in theory) and only the problem-specific details of new approaches need be implemented – abstract procedures and mechanisms for coordinating the execution of the relevant Metropolis step are already in place.

In the absence of a closed-form full conditional for the whole  $\beta$  vector, or some other appropriate multivariate proposal distribution, we would naturally tend to update only those coefficients that are directly involved in any proposed moves (to simplify the required computations and, hopefully, increase the chances of accepting the new state). In such cases — in mixture modelling for example (Richardson and Green, 1997) — the *univariate* full conditional distributions of the relevant coefficients may be available in closed form and these, of course, are potentially useful for performing the required updates. They may also greatly facilitate the ‘Gibbs sweep’ over elements of  $\beta$  that is required in ‘do nothing’ moves to ensure adequate mixing of the Markov chain (see Section 3.2.4).

#### 4. Implementation issues.

4.1. *Variable dimension vectors.* One of the most obvious difficulties in implementing reversible jump techniques is that it necessitates the use of variable-dimension parameter vectors. This raises two major issues. First, how do we store the information contained in these vectors? And second, how do we interpret that stored information? These are of particular relevance when dealing with  $\theta$  and  $\beta$  respectively and are discussed below. Before continuing, however, we first point out an important fact. To represent  $\beta$  and  $\theta$  internally as vectors of truly varying dimension would require a vast degree of dynamic memory allocation, which would impart massive overheads on memory management (even with so-called “Garbage Collection” (Jones and Lins, 1996) in place) and the result would be a much slower sampler for the vast majority of real-life applications. Hence we opt to represent these quantities internally using vectors of maximal, but fixed, size. For most of run-time there will thus exist a proportion of each vector that is entirely redundant.

We have already pointed out that without a large amount of post-processing the  $\beta$  vector is uninterpretable. With all elements except, perhaps, an ‘intercept’ term ( $\beta_1$ ) playing a potentially different role from iteration to iteration, we tend to view  $\beta$  merely as a tool for exploring parameter space rather than an object of inference. Even if we conceived of a convention for representing the redundant elements unambiguously, there would still be much scope for misinterpretation. Hence we hide the  $\beta$  vector from the user: its presence in the model is *implied* through the specification of  $\psi$  but, as far as the user is concerned, it is anonymous and cannot be accessed (directly). The implications of this are discussed below.

The  $\theta$  vector represents the configuration of the currently selected model and is more directly interpretable than  $\beta$ . It can still be difficult to examine  $\theta$  across multiple iterations but, given  $k$ , at least the meaning of each single state can be understood directly, whereas  $\beta$  can only be interpreted once the corresponding value of  $\theta$  is known. There is still much scope for misinterpretation but perhaps not enough to justify hiding  $\theta$  from the user as well. However, there are other motives for ‘internalizing’  $\theta$ . One such motive is to restrict the user’s control over the prior. If the distribution of  $\theta$  is implied by the specification of  $\psi$ , rather than being specified directly by the user, then we can always ensure that an appropriate specialized prior is selected for the problem at hand. Another reason for hiding  $\theta$  from the user is to prevent them monitoring it (i.e. storing posterior samples for it) in its standard form, which would consume unnecessarily large amounts of (storage) memory. To see why this might happen, note that in variable selection, for example, the non-redundant part of  $\theta$  represents a discrete list of currently

selected entities from a finite set. The internal size (in WinBUGS) of  $\theta$  must be the same as the size of this set in order to cover all possibilities. Bearing in mind that WinBUGS stores all parameter values, in RAM, as 32-bit (4 byte) real numbers, regardless of their ‘conceptual type’, then for a variable selection problem involving 60 covariates, say, this represents 1 MB of RAM consumption for every  $\sim 4400$  iterations (with only one chain). One million iterations would therefore require nearly 230 MB of RAM, just to store  $\theta$ , and we would still likely need to post-process the samples in order to make any meaningful inferences. Two hundred and thirty megabytes may not seem like a huge amount of RAM today (although it was beyond the reach of most computers when this work began), but this can easily be reduced by a factor of around 20 if we actually consider the information content of  $\theta$ . Recall that  $\theta$  represents a selection of  $k$  items from a total of  $Q$  items, say, and it has length  $Q$ . We may as well express this as a list of  $Q$  binary digits, with ones and zeros corresponding to selected and non-selected items respectively. Theoretically, we only need  $Q$  bits ( $Q/8$  bytes) to store this information, but we are constrained by the fact that we need to package this into 32-bit reals. Note, however, that it is possible to accurately recover 24-bit integers from 32-bit real numbers, which can subsequently be converted into 24 binary digits. Hence we can segment the binary representation of  $\theta$  into blocks of 24 and store one 32-bit real to represent each block (we actually use a ‘block-size’ of 20 in practice as it makes things simpler for the user).

Generally speaking we hide  $\theta$  from the user, but for cases in which it represents a ‘discrete selection’ we allow it to be monitored via the efficient method described above – see the examples in Sections 5 and 7 for details. For other types of model, alternative ways of accessing  $\theta$  indirectly are available. As for  $\beta$ , the implications of internalizing  $\theta$  are discussed below.

*4.2. Implications of internalizing  $\beta$  and  $\theta$ .* It is difficult to imagine a situation where the posterior distribution of  $\theta$  is of no interest at all; after all, one of the main reasons for applying a reversible jump approach is to learn about which model structures are most consistent with the observed data. Although our approach to storing this distribution for ‘discrete selections’ is efficient, it is largely useless in terms of providing the user with interpretable output: for a 20-covariate variable selection problem, for example, two models that differ by only one covariate may be stored via identifiers as far apart as  $2^{19}$  ( $\approx 500,000$ ) or as close together as  $2^0 = 1$  (or anywhere in between). Hence we provide a specialized interface for converting this type of output into something more intuitive. This comes in the form of an addi-

tional menu dedicated to processing this type of information. Currently the user can generate tables of posterior model probabilities and various graphical representations of  $\theta$  versus discrete (iteration) time, but, importantly, the interface is *extensible* (Lunn et al., 2000) and can be upgraded trivially as we learn more about users' requirements.

In cases where  $\theta$  takes on a different form, for example, if it is continuous, then we still hide it from the user as discussed above. However, we must then provide a mechanism whereby inferences about  $\theta$  can easily be drawn; again, the extensible nature of the WinBUGS framework proves invaluable here. Such mechanisms are highly context-dependant, but it will always be possible to incorporate new elements into the BUGS language to provide access to alternative representations of  $\theta$ , and to introduce new menus, etc., dedicated to summarizing them in an intuitive fashion. The same, of course, also applies to  $\beta$ , but it is far less likely that inference will be centred directly on  $\beta$  and so we have not yet implemented any specialized interfaces for summarizing its posterior. A much more likely scenario is that inference will be based, to some extent, on  $\psi' = \psi(\beta, \theta, k, X')$  where  $X'$  represents 'new' input data (e.g. new covariate values). For this reason, all instances of  $\psi$  can have analogous 'predictor nodes' coupled to them, i.e.  $\psi_p = \psi_p(\psi(\cdot), X') = \psi(\beta, \theta, k, X')$ . Representing  $\psi_p$  in terms of  $X'$  and the functional form of  $\psi$  allows  $\psi_p$  to gain access to  $\beta$  and  $\theta$  whilst keeping them hidden from the user. Note that in simple regression problems where there is only one coefficient per variable (such as in variable selection) such predictor nodes allow the posterior distribution of  $\beta$  to be probed directly, by choosing appropriate values for  $X'$  (see the example in Section 7).

Another major consequence of internalizing  $\beta$  and  $\theta$  is that it significantly limits the user's control over their prior specifications. This is not really a problem for  $\beta$  as a multivariate normal prior is both a natural choice (for coefficients) and provides the required conjugacy. In theory the mean vector and covariance matrix could be specified via the  $\psi$  function (since  $\beta$  and  $\theta$  are not hidden from  $\psi$ ). However, the dynamic nature of  $\beta$  — the fact that each element can play different roles from one iteration to the next — necessitates a homogeneous specification, i.e. the same mean and precision for each element (it is of course feasible that a degree of prior correlation might be desirable but the distribution must be invariant to a reordering of the elements). Since a mean of zero is the natural choice, the facility to specify a single precision parameter (logical or stochastic) via  $\psi$  provides the user with almost maximal permissible flexibility regarding the prior for  $\beta$ .

The prior for  $\theta$  is implicit in the choice of  $\psi$ : different  $\psi$ -functions exist

for different forms of  $\theta$ . Typically, all possible values of  $\theta$  for a given value of  $k$  are equally likely *a priori*, which is often a natural choice. Note, however, that the user has full control over the prior specification for  $k$  as it has fixed dimension and is included explicitly in the model specification. Thus adjustments can be made to the joint prior  $p(\theta, k)$  by modifying  $p(k)$  accordingly. For example,  $p(k) \propto \nu_{\theta|k}$ , where  $\nu_{\theta|k}$  is the number of possible values of  $\theta$  given  $k$  (for discrete  $\theta$ ), gives  $p(\theta, k) \propto 1$ , i.e. all values of  $\theta$  equally likely, regardless of  $k$ . In the case of variable selection, this can be achieved via a Binomial( $Q, \frac{1}{2}$ ) prior for  $k$ , which makes all  $2^Q$  possible models equally likely *a priori*:

$$p(\theta, k) = p(\theta|k)p(k) = \binom{Q}{k}^{-1} \times \binom{Q}{k} \left(\frac{1}{2}\right)^Q = \frac{1}{2^Q}$$

(See the examples in Sections 5 and 7.)

**5. Variable selection: HALD data.** In this section we analyse the ‘‘HALD’’ data presented in Draper and Smith (1981). This oft-analysed data set represents a rudimentary ‘acid test’ for variable selection techniques. We examine it here to illustrate how the BUGS language (Spiegelhalter et al., 1996a) can be used to specify variable selection problems and also to demonstrate that our implementation of the methods described herein actually works. The response variable is the amount of heat produced during the hardening of  $n = 13$  samples of cement (calories per gram). There are  $Q = 4$  potential predictor variables, which represent the percentages of the cement ingredients composed of four different compounds: (i) tricalcium aluminate; (ii) tricalcium silicate; (iii) tetracalcium alumino ferrite; and (iv) dicalcium silicate. WinBUGS code for the model described in Section 2.3 is given below. Most of the code should be self-explanatory, but several notes, which pertain to the line numbers given in the right-hand margin, are provided beneath the code for clarification. We specify the model with the multivariate-normal-gamma prior  $p(\tau, \beta|k) = \text{Ga}(\tau|a, b) \times \text{MVN}_{k+1}(\beta|\mu, \tau^{-1}\Lambda)$  where  $\Lambda = \lambda I_{k+1}$  and all elements of  $\mu$  are set to zero. This is achieved by simply specifying, via the  $\psi$ -function, a single prior precision of  $\tau\lambda^{-1}$  for all elements of  $\beta$ , and  $\tau \sim \text{Ga}(a, b)$ . The multivariate-normal-gamma prior allows us to compare our results with theoretical values.

```

model { #1
  for (i in 1:n) { #2
    Z[i] ~ dnorm(psi[i], tau) #3
  } #4
  psi[1:n] <- jump.lin.pred(X[1:n, 1:Q], k, beta.prec) #5
  id <- jump.model.id(psi[1:n]) #6
  beta.prec <- tau / lambda #7
  tau ~ dgamma(a, b) #8
  k ~ dbin(0.5, Q) #9
} #10

```

Line #5: `jump.lin.pred(.)` is the BUGS-language name for the generic linear predictor to be used for standard variable selection problems. It is a function of: (i) the full covariate matrix  $X[.,.]$ , which contains  $n$  observations on each of  $Q$  variables; (ii) the number of currently selected covariates  $k$  (the dimension of the trans-dimensional sub-model); and (iii) `beta.prec`, the prior precision to be assigned to all elements of  $\beta$ . The existence of  $\beta$  and  $\theta$  (and  $d_\beta$ ) is *implied* by the specification of a `jump.lin.pred(.)` vector; graphical nodes to represent these variables are constructed internally but are not (directly) visible to the user.

Line #6: For all trans-dimensional models in which  $\theta$  represents a ‘discrete selection’, the `jump.model.id(.)` function converts  $\theta$  into a sequence of 20-bit integers ( $\in \{0, 1, 2, \dots, 2^{20} - 1\}$ ) for efficient storage (as discussed in Section 4.1). The length of this sequence is given by  $\lceil Q/20 \rceil$ , where  $Q$  is the total number of selectable entities and  $\lceil x \rceil$  denotes the ‘ceiling’ function – the smallest integer not less than  $x$ . For this example,  $Q = 4$  and so only one scalar variable (`id`) is required to store the model configuration. The function  $\psi$  is passed as an argument to `jump.model.id(.)` instead of the more natural argument  $\theta$  because the latter is inaccessible to the user.

Lines #7 & #8: The values of `lambda`, `a` and `b` are specified in a separate ‘data file’ that it is loaded into the software after the model has been ‘declared’. For this example, a range of different values for `lambda` are specified, as discussed below, along with `a = b = 0.001`.

In Table 1 we present point estimates for the four highest posterior model probabilities from each of four analyses conducted in WinBUGS. The first three analyses are characterized by choices of  $\lambda = 100$ ,  $\lambda = 1000$ , and  $\lambda = 10000$ , whereas in the fourth analysis we set  $\lambda = 10000\tau$ , such that the prior precision for  $\beta$  becomes 10000, which is independent of  $\tau$ . For the

Prior	$\Pr(\theta = \{1, 2\} \mathcal{D})$	$\Pr(\theta = \{1, 4\} \mathcal{D})$	$\Pr(\theta = \{1, 2, 3\} \mathcal{D})$	$\Pr(\theta = \{1, 2, 4\} \mathcal{D})$
$\lambda = 10^2$	0.9589 (0.9592)	0.0075 (0.0069)	0.0253 (0.0257)	0.0062 (0.0062)
$\lambda = 10^3$	0.8946 (0.8946)	0.0900 (0.0906)	0.0081 (0.0075)	0.0048 (0.0051)
$\lambda = 10^4$	0.8519 (0.8514)	0.1418 (0.1421)	0.0024 (0.0023)	0.0020 (0.0021)
$\lambda = 10^4\tau$	0.8403	0.1482	0.0042	0.0035

Table 1: Posterior model probabilities from four separate ‘variable selection’ analyses performed on the “HALD” data set using WinBUGS (see text for details). True values, where known, are given in parentheses. Only the four most probable models are shown in each case.

first three analyses the true posterior probabilities are shown in parentheses; our reversible jump estimates are in very good agreement with these. Each estimate is based on the final half million samples from a one million iteration analysis, which took 66 seconds on a 3GHz machine.

**6. Curve fitting: Bacterial versus viral meningitis.** In this section we use a spline to model the relationship between age at diagnosis and the probability of acute bacterial meningitis (ABM) as opposed to acute viral meningitis (AVM). The model we fit belongs to the class developed and popularized by Hastie and Tibshirani (1990), known as *generalized additive models* (GAMs). GAMs are regression models where the relationships between some or all of the predictors and the dependent variable are represented by non-parametric functions (usually splines) rather than single coefficients (which avoids the potentially strong assumptions implicit in standard parametric regression). A variety of software packages are capable of maximum likelihood estimation of GAMs. However, an advantage of using the Bayesian trans-dimensional approach is that the smoothness of the fitted curve (governed by the number of ‘knots’ in the spline function – see below) is estimated as part of the model, rather than being specified in advance as is required in standard GAM-fitting algorithms.

Here we consider a GAM with a single predictor (age) although the model is easily extended to multiple explanatory variables, each with their own spline function (see the discussion in Section 8). The data correspond to 420 cases of acute meningitis from Duke University Medical Center (Spanos et al., 1989). Let  $Y_i$  represent the diagnosis for patient  $i$  ( $i = 1, \dots, n = 420$ ):  $Y_i = 1$  if patient  $i$  was diagnosed with ABM;  $Y_i = 0$  otherwise. As the response variable is binary, a natural model is

$$(6.1) \quad Y_i \sim \text{Bernoulli}(p_i); \quad \ell(p_i) = \psi(\beta, \theta, k, X_i)$$

where  $\ell(\cdot)$  denotes an appropriate ‘link’ function, typically  $\text{logit}(\cdot)$  or  $\text{probit}(\cdot)$ ,  $\psi(\cdot)$  is the spline function to be fitted, and  $X_i$  in this example denotes

the age of patient  $i$  at the time of diagnosis. Unfortunately, except for trivial definitions of  $\psi(\cdot)$ , there does not exist a prior for  $\beta$  that combines with the above likelihood to produce a closed-form full conditional for  $\beta$ . Thus, at first glance, one might think this precludes the use of our approach for fitting trans-dimensional sub-models. However, for *probit*-based linear regressions we can circumvent this lack of conjugacy by introducing auxiliary variables (Albert and Chib, 1995), which, collectively, form the  $Z$  node in our graphical model (see Figure 1). The following specification is exactly equivalent to that given in (6.1) above with  $\ell(\cdot) = \text{probit}(\cdot)$ :

$$(6.2) \quad Y_i \sim \text{Bernoulli}(p_i); \quad p_i = \begin{cases} 1 & \text{if } Z_i \geq 0 \\ 0 & \text{if } Z_i < 0 \end{cases}; \quad Z_i \sim N(\psi(\beta, \theta, k, X_i), 1)$$

To see this note that  $\Pr(Y_i = 1) = \int_0^\infty p(Z_i | \psi_i) dZ_i = 1 - \Phi(-\psi_i) = \Phi(\psi_i)$ , where  $\Phi(\cdot)$  denotes the cumulative distribution function of the standard normal distribution and  $\psi_i = \psi(\beta, \theta, k, X_i)$ . With the model specification given by (6.2) the likelihood for  $\beta$  is a product of normals, and so for cases in which  $\psi$  is linear in  $\beta$ , a multivariate normal prior for  $\beta$  leads to a multivariate normal full conditional. (Note that this method is easily extended to situations in which the response variable is ‘ordered categorical’ as opposed to binary.)

The type of spline model that we use in order to perform ‘automatic’ curve fitting is known as a *piecewise polynomial*, which is defined in terms of an unknown number of *knots*. A ‘knot’ is a point on the  $x$ -axis where the nature of the function changes; the fact that the number of knots is unknown makes this a trans-dimensional problem. In its most general form the piecewise polynomial is given by

$$\psi_i = \psi(\beta, \theta, k, X_i) = \beta_1 + \sum_{j=1}^{ord} \beta_{j+1} (X_i - x_0)^j + \sum_{j=1}^k \sum_{l=cont}^{ord} \beta_{\eta(j,l)} (X_i - \theta_j)_+^l$$

for  $X_i \geq x_0$

where:  $x_0$  is the ‘left-hand boundary’ beyond which the function is undefined, i.e.  $\psi_i = 0$  for  $X_i < x_0$ ;  $k$  is the number of knots;  $\theta_j$ ,  $j \in \{1, \dots, k\}$ , is the value of the  $j$ th knot;  $\eta(j, l) = j \times (ord - cont + 1) + l + 1$ ; and  $ord$  and  $cont$  are referred to as the ‘order’ and ‘continuity’ of the polynomial respectively. The  $x_+$  notation is defined as follows:  $x_+ = x$  if  $x > 0$ ;  $x_+ = 0$  otherwise. The order and continuity of piecewise polynomials must be such that  $ord \geq 0$  and  $0 \leq cont \leq ord$ . However, we often simply set  $cont = ord$ , which minimizes the number of terms in the polynomial such that the distance between the evaluation point ( $X_i$ ) and each knot ( $\theta_j$ ) is always raised to the same power,

i.e. *ord*. For the meningitis data in this example we fit a ‘linear’ spline with  $cont = ord = 1$ , as does Harrell (2001) for a similar data set:

$$\psi_i = \beta_1 + \beta_2(X_i - x_0) + \sum_{j=1}^k \beta_{j+2}(X_i - \theta_j)_+ \quad \text{for } X_i \geq x_0$$

Our prior for  $\beta$  is a  $k + 2$  dimensional multivariate normal distribution with zero mean and covariance matrix given by  $10^4 I_{k+2}$ . There are several options available for specifying the prior distribution for  $\theta$ . In this example, we assume each knot to be uniformly distributed on the interval  $(x_0, x_r)$  where  $x_0$  is the left-hand boundary defined above and  $x_r$  is some pre-specified maximum permissible value for all knots. Hence  $p(\theta|k) = (x_r - x_0)^{-k}$ . As there is little *a priori* information regarding the value of  $k$  we choose to illustrate the use of a hierarchical prior for this parameter. First we assume a Poisson prior with mean  $\kappa$ . However, we do not wish to allow the number of knots to exceed 20 and so we truncate this prior accordingly:  $p(k) \propto \text{Poisson}(\kappa)I(k \in \{0, 1, \dots, 20\})$ . Uncertainty about the most appropriate value for  $\kappa$  is expressed, naturally, via a further prior distribution. For this example we choose a conjugate Gamma( $\frac{9}{4}, \frac{3}{4}$ ) distribution, which specifies a prior mean and variance for  $\kappa$  of 3 and 4, respectively.

Annotated WinBUGS code for this example is presented in the appendix. Three hundred thousand iterations of a two-chain analysis, where one chain was initialized with zero knots and the other was initialized with 20 knots chosen randomly from the interval  $(x_0, x_r)$ , took around 20 minutes on a 3GHz machine. (Note that the values specified for  $x_0$  and  $x_r$  were 0 and 75 respectively.) The posterior distribution of the model fit, which is based on 400,000 posterior samples from iterations 100001–300000, is summarised in Figure 2. The model is in good agreement with the plotted points, although note that these are empirical probabilities calculated by arbitrarily grouping the raw data, rather than individual observations. Convergence was assessed in this case by considering the similarity of the two Markov chains for a random subset of the 420  $\psi_i$ s. Standard fixed-dimension convergence diagnostics are applicable to problems of this type as the main object of inference, the model fit, has fixed dimension.

**7. Variable selection: Pharmacogenetic effects of variants of the APOE gene.** This final example illustrates both the performance of our variable selection approach for a ‘real’ data analysis and, also, how one might draw inferences on the hidden vector  $\beta$ . The response variable in this case is the reduction in LDL-cholesterol in 327 patients following treatment with atorvastatin at a dosage of 10mg/day for 52 weeks.

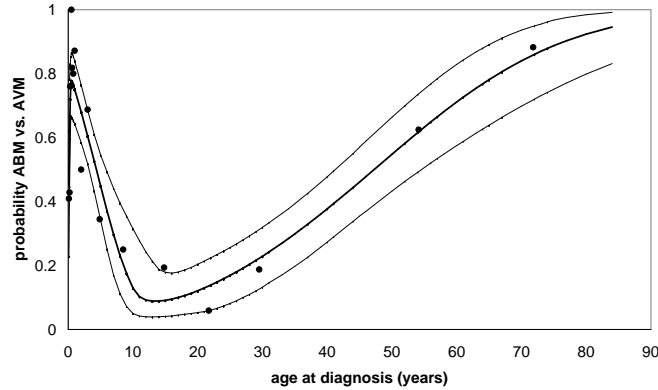


Figure 2: Posterior median (with 95% credible interval) fitted relationship between age at diagnosis and probability of acute bacterial meningitis (ABM) as opposed to acute viral meningitis (AVM). The fit is based on 420 observed binary diagnoses; the plotted points ( $\bullet$ ) are empirical probabilities calculated from the raw data.

There is a wealth of literature discussing the association between variants of the APOE gene and the metabolic regulation of cholesterol as well as Alzheimer’s disease (Eichner et al., 2002; Puglielli et al., 2003; Tanzi and Bertram, 2001). For this example the set of available covariates comprises gender, body mass index, age, and patient-specific genotypes assayed at twelve distinct SNP marker loci on the APOE gene. (SNP, pronounced ‘snip’, stands for *Single Nucleotide Polymorphism*. A SNP marker locus is a position on the human genome where a single nucleotide base (A, C, G or T) is known to exhibit non-negligible ( $> 5\%$ , say) variation across the population.) Current SNP assays simply measure the number of copies of a specific ‘allele’ at a given locus (the assay is designed to detect amounts of A, C, G or T) and so the resulting covariate is categorical with values in  $\{0, 1, 2\}$  (since humans have two chromosomes). It is of course feasible that one or more SNPs are in some way causative of a pharmacogenetic effect on patients’ responses to atorvastatin treatment, and this of course would be of great interest if detectable. However, non-causative SNPs in close proximity on the genome to causal mutations may also be associated with differences in response to treatment due to the fact that, in general, the probability that a given section of genetic material is transmitted intact, from one generation to the next, is inversely related to the length of that section. (See

$\theta$	posterior probability	cumulative probability
{9}	0.125	0.125
{9, 23}	0.0796	0.204
{9, 13}	0.0680	0.272
{3, 9}	0.0163	0.289
{9, 22}	0.0149	0.303
{9, 24}	0.0135	0.317
{9, 15}	0.0135	0.330
{9, 13, 23}	0.0133	0.344
intercept only	0.0129	0.357
{9, 22, 23}	0.0116	0.368
{8, 9}	0.0106	0.379

Table 2: Posterior model probabilities from WinBUGS analysis of APOE data (see text for details). Only those models for which the posterior probability  $> 0.01$  are shown.

Thomas (2004) for a detailed discussion of so-called ‘association studies’.) Hence a variable selection analysis regressing response on the various SNP markers can reveal important information about the location of any causal mutations. A reversible jump approach avoids problems of multiple testing as well as allowing complex multivariate associations to be detected.

One obvious approach to dealing with these data is to directly include the 12 assayed genotypes —  $g_{ij} \in \{0, 1, 2\}$ ,  $i = 1, \dots, n = 327$ ,  $j = 1, \dots, 12$  — in the covariate matrix  $X$  and run the same type of analysis as previously discussed. However, this would be somewhat naive as genetic effects often do not manifest themselves in a ‘dose-dependent’ way. Instead we decompose each  $g_{ij}$  into two indicator variables,  $I(g_{ij} = 0)$  and  $I(g_{ij} = 2)$ , and incorporate both of these into  $X$ . Selecting either variable alone represents a dominant/recessive type of effect whereas various levels of ‘co-dominance’ (where the effect of one allele lies somewhere between the effects of zero and two alleles) can be accounted for by selecting both indicators together.

For our analysis the  $X$  matrix is arranged as follows: columns 1–12 are given by  $I(g_{.j} = 0)$ ,  $j = 1, \dots, 12$ ; columns 13–24 are given by  $I(g_{.j} = 2)$ ,  $j = 1, \dots, 12$ ; and columns 25–27 represent age, body mass index, and gender, respectively. Annotated WinBUGS code is presented in the appendix and all models whose posterior probability was greater than one per cent are shown, in rank order, in Table 2. The results are based on 500,000 posterior samples from the final 250,000 out of 500,000 iterations of a two-chain analysis, which took 34 minutes on a 3GHz machine. The two chains

differed in their initial states for  $\theta$  and  $k$ : one chain was started at the null model (intercept only,  $\theta = \emptyset$ ,  $k = 0$ ) and the other was started at the saturated model ( $\theta = \{1, \dots, Q\}$ ,  $k = Q = 27$ ). As with fixed-dimension models, running multiple chains from ‘over-dispersed’ starting values can aid greatly in the detection of convergence. When two chains that have been initialised so far apart start spending much of their time in the same state (with respect to  $\{\theta, k\}$ ), our intuition suggests that convergence should be reasonably close. More formally, convergence diagnosis in these cases should include, at the very least, some assessment of how similar are the tables of posterior model probabilities generated by each chain. The above ‘burn in’ is greatly conservative in this respect, partly because diagnostic tools for between chain comparisons have not yet been fully implemented within the software. Visual inspection of trace plots of model state versus iteration number suggest that convergence actually occurs within the first 10,000 iterations.

One of the first things to notice from Table 2 is that there is no strong signal in these data. The most probable model under the posterior distribution represents only 12.5 per cent of that distribution and the 11 models shown in Table 2 cover less than 40 per cent. Indeed, over 16000 distinct models are visited during the MCMC simulation! Note that these are *accepted* models, that is they must be reasonably well supported by the data (and prior). Whilst there appears to be no particularly dominant single model, there certainly does appear to exist an important marker locus. Variable number 9, which corresponds to zero copies of the C nucleotide (as opposed to T) at the locus known to us as ‘L4870’, appears in 10 out of the 11 models shown in Table 2 and contributes to over 90 per cent of all accepted models. This latter figure, 0.906 to be precise, represents the variable’s *marginal* posterior probability. It is interesting to note that variable number 21, the other indicator associated with locus L4870, is not present in the top 11 models. Indeed, less than four per cent of all accepted models contain variable number 21, and so there is little evidence of any difference between the ‘effects’ of one and two C nucleotides at this locus. Of course this is not necessarily indicative of a recessive/dominant effect at the causal locus – if the association isn’t particularly strong there may be little additional benefit in using an extra degree of freedom to model any ‘dose-dependence’.

Figure 3 shows the marginal posterior probabilities for all variables included in our analysis. These are calculated automatically by the software but can also be calculated via the model specification as illustrated in the appendix. With  $\theta$  inaccessible to the user one of the easiest ways in which to determine whether a given variable is included in the model at a given time is

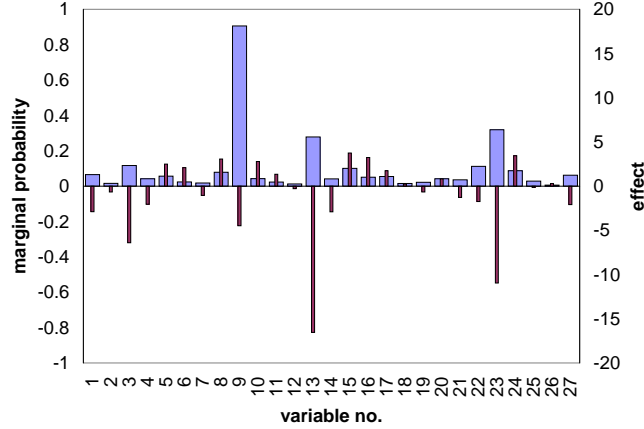


Figure 3: Marginal posterior probabilities and effect sizes associated with each covariate included in APOE data analysis. Marginal probabilities are shown as wide bars and are measured on the left-hand  $y$ -axis. Posterior means for the effect of each variable conditional on it being included in the model are shown as narrow/dark bars – these are measured on the right-hand  $y$ -axis.

via prediction of  $\psi' = \psi(\beta, \theta, k, X')$ . Let  $\psi'_i = \psi(\beta, \theta, k, X'_i)$ ,  $i = 1, \dots, Q + 1$ , where  $X'_i$  represents a set of new covariate values at which the linear predictor is to be evaluated and is given by the  $i$ th row of

$$X' = \begin{pmatrix} I_Q \\ 0_Q \end{pmatrix}$$

Here  $I_Q$  is the  $Q \times Q$  identity matrix and  $0_Q$  is a  $1 \times Q$  row-vector of zeros. If we define  $e_i = \psi'_i - \psi'_{Q+1}$ ,  $i = 1, \dots, Q$ , then these  $e_i$  terms represent the posterior ‘effects’ of each individual variable. We can derive the marginal posterior probability of variable  $i$  being included in the model,  $\Pr(i \in \theta | \mathcal{D})$ , by calculating the posterior mean of  $I(|e_i| > 0)$ . Similar calculations will give us the posterior probability associated with a specific *group* of variables of interest. For example, we may be interested in the probability of a specific genetic locus being involved in the model, e.g.  $\Pr(\{i \in \theta\} \cup \{i + 12 \in \theta\} | \mathcal{D})$  for some  $i \in \{1, \dots, 12\}$ . Of course the  $e_i$  terms are of interest themselves, but it is their values conditional on  $i \in \theta$  that are (arguably) more informative. The posterior mean of each  $e_i$ , conditional on  $i \in \theta$ , is also shown in Figure

3:

$$\mathbb{E}(e_i | i \in \theta, \mathcal{D}) = \frac{\mathbb{E}(e_i | \mathcal{D})}{\Pr(i \in \theta | \mathcal{D})}$$

The variables exhibiting the highest marginal posterior probabilities are numbers 9, 23 and 13, which also make up the top three models shown in Table 2. These variables correspond to loci ‘L4870’, ‘L7327’ and ‘L650’ respectively. (Note that their counterparts, variables 21, 11 and 1, are all selected infrequently.) In each case the associated effect is negative, but as the selected variables correspond to genotypes of 0, 2, and 2 copies of the rarer nucleotide, respectively, the nature of the effects is somewhat different. In the case of L4870, individuals homozygous for the more common (‘wild-type’) nucleotide show a better response to treatment than individuals who are heterozygotes or homozygous for the rarer nucleotide. In contrast, for loci L650 and L7327, individuals who are homozygous for the rarer allele show a better response to atorvastatin than others. It is interesting to note that effects of L4870 on baseline cholesterol, and on response to ‘statin’ treatment, have been reported elsewhere (Ordovas, 2004).

**8. Discussion.** We have identified a generic class of trans-dimensional models for which posterior samples can be generated straightforwardly using reversible jump MCMC. Our method requires little more than the ability to derive full conditional distributions in closed form (when such closed forms exist) and so comes at little additional cost to that required to analyse an analogous fixed-dimension model via Gibbs sampling. We have implemented our approach using the popular WinBUGS framework as a sampling engine. The reasons for this are many but among the most important are: (i) most importantly of all, WinBUGS is a very general-purpose piece of software and as such can accommodate our trans-dimensional models within arbitrary other models (e.g. hierarchical), meaning that they may be used in virtually any setting, regardless of whether or not the ‘response variable’ or the ‘input data’ are observed, and irrespective of any associated sub-models; (ii) WinBUGS was designed to derive full conditional distributions in closed form wherever possible, and so the software naturally provides some of the fundamental techniques required for our approach; (iii) the software is now very widely used, and so many researchers will be able to use our approach with a minimal learning investment.

Of course, there is no reason why two or more trans-dimensional sub-graphs cannot be incorporated into the same full probability model using our approach. For example, in cases where the response of interest is multivariate we may wish to adopt a *Seemingly Unrelated Regressions* (SUR)

approach (Dennison et al., 2002; Verzilli et al., 2005). Here a separate variable selection model is assumed for the mean of each response variable and any ‘unexplainable’ correlation between the response variables is accounted for via an assumption of multivariate normality. Another example entails fitting a separate piecewise polynomial to each of a number of explanatory variables for a single response variable (this is a type of GAM model). In this case it is necessary (for identifiability) to remove the intercept term from each spline and incorporate a single intercept into the fixed dimension part of the model instead. It is also possible to impose the constraint that multiple trans-dimensional sub-models have the same dimension parameter,  $k$ , so long as the coefficient vectors ( $\beta_i$ , say) for each sub-model have distinct *Markov blankets* (the Markov blanket of a node  $u$  is the set of nodes on which  $u$  depends in its full conditional distribution, i.e. its parents, its children, and all co-parents of its children). (Situations in which two or more trans-dimensional sub-models share common  $k$  and  $\theta$  parameters are actually single trans-dimensional models and can be specified instead via an appropriate definition of  $\psi$ .)

Other elaborations of the models described in this paper include the following. Regressions may be robustified against the influence of outliers by making Student- $t$  assumptions regarding the distribution of the response variable, for example.  $FC(\beta)$  will still be available in closed form because, with the aid of an auxiliary variable, the Student- $t$  distribution can be expressed as a normal distribution (Johnson and Kotz, 1972; Wakefield et al., 1994). The ‘degrees of freedom’ parameter may be specified as known or assigned an appropriate prior and ‘estimated’. (Note that the same trick could be used to specify a multivariate- $t$  prior for  $\beta$ .) Another useful elaboration is the specification of a hierarchical prior for  $k$  (as in Section 6) and/or the precision of  $\beta$ . Normally these would be hyperparameters, with fixed prior parameters. However, in order to reduce any sensitivity to these prior parameters, it may be desirable to express some uncertainty regarding their values.

The WinBUGS framework has been constructed according to a *component-oriented* philosophy (Szyperski, 1995). This novel software engineering approach aims to create fully extensible, modular systems. Software is composed of a number of components that are not linked together until load-time or even run-time. Inclusion of new methods and applications is achieved by writing extra components that simply either ‘plug in’ to relevant slots in existing components, or make use of existing components, without requiring any part of the software to be modified or even recompiled. All of this means that the software can evolve rapidly. As new model

types are implemented, they may be distributed immediately in ‘patch’ (or plug-in) form, along with any new (specialized) interfaces that are required for drawing inferences on  $\beta$  and  $\theta$  as a consequence of these vectors being ‘internalized’ (as discussed in Section 4). Users of the software may also play a more interactive role in the development of new ideas, with a relatively quick turnaround being possible between conception (of a new model type or specialized interface, say) and implementation.

Convergence diagnosis remains a key issue with trans-dimensional models. For situations in which  $\theta$  is discrete, it seems logical to compare tables of estimated posterior model probabilities from multiple chains initialized in ‘over-dispersed’ states. For cases in which  $\theta$  is continuous but not a central object of inference, such as in spline-fitting, standard convergence diagnostics (Cowles and Carlin, 1996; Mengersen et al., 1999) applied to appropriate quantities in the fixed-dimension part of the model should suffice. More generally, the method proposed by Brooks and Giudici (1999) and subsequently extended by Castelloe and Zimmerman (2002) looks very promising. Here the two-way ANOVA originally proposed by Gelman and Rubin (1992) is extended to a three-way ANOVA in which the additional ‘level’ corresponds to ‘model state’. A version of the original (two-way) approach (Brooks and Gelman, 1998; Spiegelhalter et al., 2003) is already implemented in WinBUGS, but it is not yet clear to us how this might be adapted. When such a diagnostic does become available, however, the component-oriented design of WinBUGS means that it may be distributed across the user community with minimal effort. Alternatively, by making reversible jump techniques available to a wider audience we may end up being guided towards an even more practical solution to convergence diagnosis.

## APPENDIX A: WINBUGS CODE

**A.1. Meningitis data.** Annotated WinBUGS code for fitting a linear spline to the meningitis data of Section 6 is as follows.

```

model { #1
  for (i in 1:n) { #2
    Y[i] ~ dbern.aux(Z[i]) #3
    Z[i] ~ dnorm(psi[i], 1) #4
    prob[i] <- phi(psi[i]) #5
  } #6
  psi[1:n] <- jump.pw.poly.c.lin(X[1:n], k, beta.prec, #7
                                x.0, x.r) #7
  beta.prec <- 0.0001 #8
  k ~ dpois(kappa)I(, 20) #9
  kappa ~ dgamma(2.25, 0.75) #10
} #11

```

- Line #3: The `dbern.aux(.)` distribution is a Bernoulli distribution with a probability parameter that can only equal zero or one. It is parameterized in terms of a continuous variable defined on  $\mathbb{R}^1$ . If its parameter (`Z[i]` in this case) is greater than or equal to zero then the ‘success probability’ of the Bernoulli is one; a value less than zero, on the other hand, corresponds to a success probability of zero (see Equation (6.2)). Hence this distribution is equivalent to the deterministic function  $I(\text{parameter} \geq 0)$ .
- Line #5: As the specified model is equivalent to a probit regression, the underlying probability of ABM vs. AVM plotted in Figure 2 is given by  $\Phi(\psi_i)$ , where  $\Phi(.)$  is the cumulative distribution function of the standard normal distribution. The `phi(.)` function in WinBUGS returns the appropriate value of  $\Phi(.)$ .
- Line #7: `jump.pw.poly.c.lin(.)` is the BUGS syntax for a linear piecewise polynomial with ‘continuous’ knots. It is a function of: (i) `X[.]`, the vector of  $x$ -values at which the polynomial is to be evaluated; (ii) the number of knot-points `k`; (iii) `beta.prec`, the prior precision to be assigned to all elements of  $\beta$ ; (iv) the left-hand boundary `x.0`, to the left of which the function is undefined; and (v) `x.r`, the maximum permissible value for all knots (note that `x.0` is also the minimum value for all knots).

**A.2. APOE data.** Annotated WinBUGS code for performing variable selection on the APOE data of Section 7 is given below.

```

model { #1
  for (i in 1:n) { #2
    Z[i] ~ dnorm(psi[i], tau) #3
    for (j in 1:12) { #4
      X[i, j] <- equals(g[i, j], 0) #5
      X[i, (j + 12)] <- equals(g[i, j], 2) #6
    } #7
    X[i, 25] <- age[i] #8
    X[i, 26] <- bmi[i] #9
    X[i, 27] <- sex[i] #10
  } #11
  psi[1:n] <- jump.lin.pred(X[1:n, 1:Q], k, beta.prec) #12
  id[1:2] <- jump.model.id(psi[1:n]) #13
  beta.prec <- 0.0001 #14
  tau ~ dgamma(a, b) #15
  k ~ dbin(0.5, Q) #16
  pred[1:(Q + 1)] <- jump.lin.pred.pred(psi[1:n], #17
                                         X.pred[1:(Q + 1), 1:Q]
  for (i in 1:Q) { #18
    X.pred[i, i] <- 1 #19
    for (j in 1:(i - 1)) {X.pred[i, j] <- 0} #20
    for (j in (i + 1):Q) {X.pred[i, j] <- 0} #21
    X.pred[(Q + 1), i] <- 0 #22
    e[i] <- pred[i] - pred[Q + 1] #23
    marginal[i] <- step(abs(e[i]) - eps) #24
  } #25
} #26

```

Lines #5 & #6: The `equals(.,.)` function in WinBUGS returns the value one if and only if the first argument is equal to the second argument; the value zero is returned otherwise.

Line #13: In this example  $Q = 27$  and so  $\lceil 27/20 \rceil = 2$  variables are required to store the sequence of 20-bit integers returned by the `jump.model.id(.)` function.

Line #17: The `jump.lin.pred.pred(.)` syntax enables prediction based on linear predictors specified via the `jump.lin.pred(.)` function. Its arguments are: (i) the set of linear predictors on which prediction is to be based – this merely provides access to the relevant  $\beta$  and  $\theta$  vectors; and (ii) the  $Q$ -column matrix of covariate values for which predictions are required.

Line #24: The `step(.)` function in WinBUGS returns the value one if and only if

its argument is greater than or equal to zero; the value zero is returned otherwise. Hence, if `eps` is suitably small (e.g.  $10^{-20}$ ), the posterior mean of `marginal[i]` gives the marginal posterior probability of variable `i` being included in the model.

### ACKNOWLEDGEMENTS

We would like to thank Sylvia Richardson, Claudio Verzilli and James Bennett for helpful discussions.

### REFERENCES

- Albert, J. and Chib, S. (1995). Bayesian residual analysis for binary response regression models. *Biometrika*, 82:747–759.
- Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. Wiley Series in Applied Probability and Statistics. John Wiley & Sons, New York.
- Brooks, S. P. and Gelman, A. (1998). Alternative methods for monitoring convergence of iterative simulations. *J. Comput. Graph. Stat.*, 7:434–455.
- Brooks, S. P. and Giudici, P. (1999). Convergence assessment for reversible jump MCMC simulations. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 6*, pages 733–742, Oxford. Oxford University Press.
- Brooks, S. P., Giudici, P., and Roberts, G. O. (2003). Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions (with discussion). *J. Roy. Stat. Soc. B*, 65:3–55.
- Carlin, B. P. and Chib, S. (1995). Bayesian model choice via Markov chain Monte Carlo methods. *J. Roy. Stat. Soc. B*, 57:473–484.
- Castelloe, J. M. and Zimmerman, D. L. (2002). Convergence assessment for reversible jump MCMC samplers. Technical Report 313, Department of Statistics and Actuarial Science, University of Iowa.
- Cornell, G. and Horstmann, C. S. (1997). *Core Java, 2nd Edition*. Prentice Hall, New Jersey.
- Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *J. Am. Stat. Assoc.*, 91:883–904.
- Dennison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M. (2002). *Bayesian Methods for Non-linear Classification and Regression*. John Wiley & Sons, Chichester.
- Draper, N. and Smith, H. (1981). *Applied Regression Analysis, 2nd Edition*. John Wiley & Sons, New York.
- Eichner, J. E., Dunn, S. T., Perveen, G., Thompson, D. M., Stewart, K. E., and Stroehla, B. C. (2002). Apolipoprotein E polymorphism and cardiovascular disease: a HuGE review. *Am. J. Epidemiol.*, 155:487–495.
- Frühwirth-Schnatter, S. and Wagner, H. (2004). Data augmentation and Gibbs sampling for regression models of small counts. Technical report, IFAS, Johannes Kepler Universität Linz, Austria, <http://www.ifas.jku.at/>.
- Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *J. Am. Stat. Assoc.*, 85:398–409.
- Gelman, A. (2004). Prior distributions for variance parameters in hierarchical models. Technical report, Department of Statistics and Department of Political Science, Columbia University, New York, USA, <http://www.stat.columbia.edu/~gelman/>.

- Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Stat. Sci.*, 7:457–511.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE T. Pattern Anal.*, 6:721–741.
- George, E. I. and McCulloch, R. E. (1996). Stochastic search variable selection. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pages 203–214. Chapman and Hall, London.
- Gilks, W. (1992). Derivative-free adaptive rejection sampling for Gibbs sampling. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pages 641–665, Oxford. Oxford University Press.
- Gilks, W. R. (1996). Full conditional distributions. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pages 75–88. Chapman and Hall, London.
- Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732.
- Harrell, F. E. (2001). *Regression Modelling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer, London.
- Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Chapman and Hall, London.
- Hastings, W. K. (1970). Monte Carlo sampling-based methods using Markov chains and their applications. *Biometrika*, 57:97–109.
- Holmes, C. C. and Held, L. (2005). Bayesian auxiliary variable models for binary and polychotomous regression. *Bayesian Analysis*. To appear.
- Johnson, N. L. and Kotz, S. (1972). *Distributions in Statistics: Continuous Multivariate*. John Wiley & Sons, New York.
- Jones, R. and Lins, R. (1996). *Garbage Collection: Algorithms for Automatic Dynamic Memory Management*. John Wiley & Sons, New York.
- Lauritzen, S. L., Dawid, A. P., Larsen, B. N., and Leimer, H. G. (1990). Independence properties of directed Markov fields. *Networks*, 20:491–505.
- Lunn, D. J. (2003). WinBUGS Development Interface (WBDev). *ISBA Bulletin*, 10(3):10–11.
- Lunn, D. J., Best, N., Thomas, A., Wakefield, J., and Spiegelhalter, D. (2002). Bayesian analysis of population PK/PD models: general concepts and software. *Journal of Pharmacokinetics and Pharmacodynamics*, 29:271–307.
- Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. *Stat. Comput.*, 10:325–337.
- Mengersen, K. L., Robert, C. P., and Guihenneuc-Jouyaux, C. (1999). MCMC convergence diagnostics: a review. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 6*, pages 415–440, Oxford. Oxford University Press.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1091.
- Neal, R. M. (1997). Markov chain Monte Carlo methods based on ‘slicing’ the density function. Technical Report 9722, Dept. of Statistics, University of Toronto.
- Ordovas, J. M. (2004). Pharmacogenetics of lipid diseases. *Human Genomics*, 1(2):111–125.
- Puglielli, L., Tanzi, R. E., and Kovacs, D. M. (2003). Alzheimer’s disease: the cholesterol connection. *Nat. Neurosci.*, 6:345–351.
- Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). Bayesian Model Averaging for linear regression models. *J. Am. Stat. Assoc.*, 92:179–191.

- Reiser, M. and Wirth, N. (1992). *Programming in Oberon: Steps Beyond Pascal and Modula*. ACM Press, New York.
- Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *J. Roy. Stat. Soc. B*, 59:731–792.
- Sisson, S. A. (2005). Trans-dimensional Markov chains: a decade of progress and future perspectives. *J. Am. Stat. Assoc.* To appear.
- Spanos, A., Harrell, F. E., and Durack, D. T. (1989). Differential diagnosis of acute meningitis: An analysis of the predictive value of initial observations. *J. Am. Med. Assoc.*, 262:2700–2707.
- Spiegelhalter, D., Thomas, A., Best, N., and Gilks, W. (1996a). *BUGS 0.5: Bayesian inference Using Gibbs Sampling – Manual (version ii)*. Medical Research Council Biostatistics Unit, Cambridge.
- Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D. (2003). *WinBUGS User Manual, Version 1.4*. Medical Research Council Biostatistics Unit, Cambridge.
- Spiegelhalter, D. J. (1998). Bayesian graphical modelling: a case-study in monitoring health outcomes. *Appl. Stat.*, 47:115–133.
- Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., and Cowell, R. G. (1993). Bayesian analysis in expert systems (with discussion). *Stat. Sci.*, 8:219–283.
- Spiegelhalter, D. J., Thomas, A., and Best, N. G. (1996b). Computation on Bayesian graphical models. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 5*, pages 407–425, Oxford. Oxford University Press.
- Szyperski, C. (1995). Component-oriented programming: A refined variation of object-oriented programming. *The Oberon Tribune*, 1:1–5.
- Tanzi, R. E. and Bertram, L. (2001). New frontiers in Alzheimer’s disease genetics. *Neuron*, 32:181–184.
- Thomas, A., Best, N., Lunn, D., Arnold, R., and Spiegelhalter, D. (2004). *GeoBUGS User Manual: Version 1.2*. Dept. Epidemiology and Public Health, Imperial College School of Medicine, London.
- Thomas, D. C. (2004). *Statistical Methods in Genetic Epidemiology*. Oxford University Press, Oxford.
- Troughton, P. T. and Godsill, S. J. (1998). A reversible jump sampler for autoregressive time series. *Proceedings of IEEE ICASSP-98*, pages 2257–2260.
- Verzilli, C. J., Stallard, N., and Whittaker, J. C. (2005). Bayesian modelling of multivariate quantitative traits using Seemingly Unrelated Regressions. *Genet. Epidemiol.* To appear.
- Waagepetersen, R. and Sorensen, D. (2001). A tutorial on reversible jump MCMC with a view toward applications in QTL-mapping. *Int. Stat. Rev.*, 69(1):49–61.
- Wakefield, J. C., Smith, A. F. M., Racine-Poon, A., and Gelfand, A. E. (1994). Bayesian analysis of linear and non-linear population models by using the Gibbs sampler. *Appl. Stat.*, 43:201–221.
- Whittaker, J. (1990). *Graphical Models in Applied Multivariate Analysis*. John Wiley & Sons, Chichester.

DEPARTMENT OF EPIDEMIOLOGY AND PUBLIC HEALTH,  
 IMPERIAL COLLEGE LONDON,  
 ST. MARY’S CAMPUS,  
 NORFOLK PLACE,  
 LONDON W2 1PG, UK  
 E-MAIL: d.lunn@imperial.ac.uk