

Perfect Sampling for Bayesian Variable Selection in a Linear Regression Model

U. Schneider and J. N. Corcoran
University of Colorado *

October 7, 2002

Abstract

We describe the use of perfect sampling algorithms for Bayesian variable selection in a linear regression model. Starting with a basic case solved by Huang and Djurić (2002), where the model coefficients and noise variance are assumed to be known, we generalize the model step by step to allow for other sources of randomness, specifying perfect simulation algorithms that solve these cases by incorporating various techniques including Gibbs sampling, the perfect independent Metropolis- Hastings (IMH) algorithm, and recently developed “slice coupling” algorithms. Applications to simulated data sets suggest that our algorithms perform well in identifying relevant predictor variables.

1 Introduction

Variable selection is an important and well-studied problem in many areas. There are various approaches to solve this problem— among them Bayesian methods that use Markov chain Monte Carlo (MCMC) algorithms.

A major problem with conventional MCMC methods is the assessment of convergence. How long do we need to run a chain to get “good enough” samples from the target distribution? This question vanishes completely with the use of perfect simulation. Perfect simulation (“perfect sampling”, “exact sampling”, “coupling from the past” (CFTP)) is an interesting twist on MCMC methods that guarantees that the sample is an *exact draw* from the target distribution. While not applicable for *any* target distribution, perfect simulation can be implemented for a promisingly increasing class of probability distributions.

This paper is organized as follows:

*Postal Address: Department of Applied Mathematics, University of Colorado, Box 526 Boulder CO 80309-0526, USA; email: corcoran@colorado.edu, Uli.Schneider@colorado.edu; phone: 303-492-0685

⁰Keywords: invariant measures, backwards coupling, coupling from the past, exact sampling, perfect sampling, slice coupling, shift coupling, Bayesian variable selection
AMS Subject classification: 60J10, 65C05, 62F25, 60K30

In Section 2 we formulate the problem of variable selection in a linear regression model. Section 3 deals with perfect sampling methods in general, as well as describing the so-called IMH-algorithm in Section 3.1 and the idea of slice coupling in Section 3.2. We go back to variable selection in Section 4 and start off by describing Huang and Djurić's approach using perfect simulation (Huang and Djurić, 2002). We then generalize the model step by step, allowing more and/or different components to be random. We specify perfect sampling algorithms that can be used to solve these different cases by incorporating ideas from Section 3. In Section 5, we present promising results that were found testing the different algorithms on artificial data sets. At last, the Appendix contains an idea for future research that might allow a reduction the computational cost of some of the algorithms.

2 The Model

Consider data records given by a linear regression model with iid Gaussian noise

$$\mathbf{y} = \gamma_1 \theta_1 \mathbf{x}_1 + \cdots + \gamma_q \theta_q \mathbf{x}_q + \boldsymbol{\varepsilon}, \quad (1)$$

where \mathbf{y} is an $n \times 1$ response vector, $\mathbf{x}_i \in \mathbb{R}^n$ are $n \times 1$ vectors of predictors, $\theta_i \in \mathbb{R}$ are the corresponding coefficients, and $\gamma_i \in \{0, 1\}$ are indicators taking values 0 and 1 for $i = 1, \dots, q$. The noise vector $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is assumed to have independent components with $\varepsilon_j \sim N(0, \sigma^2)$ for $j = 1, \dots, n$.

Our task is to recover from the data the subset of the q predictors that are a part of the model, that is, to determine the value of the indicator vector $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_q)'$.

We approach this problem from a Bayesian perspective, selecting as an optimal $\boldsymbol{\gamma}$ the one that appears most frequently when we use perfect sampling algorithms to draw from the posterior density

$$\pi_{\Gamma, \sigma^2, \Theta | \mathbf{Y}}(\boldsymbol{\gamma}, \sigma^2, \boldsymbol{\theta} | \mathbf{y}) \propto L(\boldsymbol{\gamma}, \sigma^2, \boldsymbol{\theta}) g(\boldsymbol{\gamma}, \sigma^2, \boldsymbol{\theta}) \quad (2)$$

of the parameters given the data. Here, $L(\cdot)$ is the likelihood, $g(\cdot)$ is a prior for the parameters, and $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_q)'$.

If σ^2 and $\boldsymbol{\theta}$ were known, then we would expect that the most frequently appearing values of $\boldsymbol{\gamma}$ would coincide with the *maximum a posteriori* (or MAP) estimator which is the argument that maximizes the *a posteriori* probability

$$\pi_{\Gamma | \mathbf{Y}}(\boldsymbol{\gamma} | \mathbf{y}) \propto L(\boldsymbol{\gamma}) g(\boldsymbol{\gamma}).$$

In this case, one could perform an exhaustive search through the 2^q combinations of vectors $\boldsymbol{\gamma}$ provided q is reasonable.

The priors

We use the standard normal-gamma conjugate class of priors suggested by Raferty et al. (1997),

$$\frac{\lambda\nu}{\sigma^2} \sim \chi^2(\nu) \longrightarrow Z := \frac{1}{\sigma^2} \sim \Gamma\left(\frac{\nu}{2}, \frac{\lambda\nu}{2}\right) \quad (3)$$

$$\boldsymbol{\theta} \sim N(\boldsymbol{\xi}, \sigma^2 \mathbf{V}) \quad (4)$$

in addition to the non-informative prior for Γ :

$$\gamma_i \stackrel{iid}{\sim} \text{Bernoulli}\left(\frac{1}{2}\right), \quad i = 1 \dots, q. \quad (5)$$

Here, λ and ν and \mathbf{V} are hyperparameters to be chosen.

3 Perfect Simulation

There has been considerable recent work on the development and application of “perfect sampling” algorithms that will enable the simulation of the invariant (or stationary) measure π of a Markov chain, either exactly (that is, by drawing a random sample known to be from π) or approximately, but with computable order of accuracy. These were sparked by the seminal paper of Propp and Wilson (1996), and several variations and extensions of this idea have appeared since – see Fill (1998), Foss and Tweedie (1998), Foss et al. (1998), Häggström et al. (1999), Kendall (1998), Møller (1999), and Murdoch and Green (1998). These ideas have proven effective in areas such as statistical physics, spatial point processes and operations research, where they provide simple and powerful alternatives to methods based on iterating transition laws, for example.

The essential idea of most of these approaches is to find a random epoch $-T$ in the past such that, if we construct sample paths (according to a transition law $P(x, y)$ that is invariant for π) from every point in the state space starting at $-T$, then all paths will have coupled successfully by time zero. The common value of the paths at time zero is a draw from π . Intuitively, it is clear why this result holds with such a random time T . For consider a chain starting at $-\infty$ with the stationary distribution π . At every iteration it maintains the distribution π . But at time $-T$ it must pick *some* value x , and from then on it follows the trajectory from that value. But of course it arrives at the same place at time zero no matter what value x is picked at time $-T$: so the value returned by the algorithm at time zero must itself be a draw from π .

Perfect sampling algorithms can be particularly efficient if the chain is *stochastically monotone* in the sense that paths from lower starting points stay below paths from higher starting points. In this case, one need only couple sample paths from the “top” and “bottom” of the space, as all other paths will be sandwiched in between. It is possible to generalize one step further to monotone chains on an unbounded state space by considering *stochastically dominating* processes to bound the journeys of sample paths.

For this to be practicable we need to ensure that T is indeed finite. Propp and Wilson (1996) show that this occurs for irreducible aperiodic finite space chains, and for a number of stochastically monotone chains possessing maximal and minimal elements. Indeed, Corcoran and Tweedie (2002) show that if the distribution at $-\infty$ is any fixed (or even random) value x_0 , then under fairly standard conditions the value at time zero is still distributed according to π , and this observation is crucial in what follows.

There are several easy-to-read perfect sampling tutorials available, and we refer interested readers to Casella et al. (2000). In this paper we only wish to emphasize that the key idea in the search successively further and further back in time for the so-called *backward coupling time* T requires that one reuse random number streams. That is, if sample paths run forward to time 0 from time -1 using a random number (or random vector) U_{-1} have not coalesced by time 0, then one must go back further, say to time -2 and run paths forward for two steps using a random number U_{-2} and then the **previously used** U_{-1} .

We now describe two specific perfect sampling schemes that will be used in our model selection algorithms.

3.1 The IMH Algorithm

Certain monotonicity properties of the “independent” Metropolis-Hastings (IMH) scheme, which we review briefly in this section, are such that perfect sampling is feasible (Corcoran and Tweedie, 2002). We use the term “independent” to describe the Metropolis-Hastings algorithm where candidate states are generated by a distribution that is independent of the current state of the chain. In other words, suppose we have a given candidate distribution Q which we will assume to have a density q , positive everywhere for convenience, with which we can generate potential values of an i.i.d. sequence. A “candidate value” generated according to q is then accepted with probability $\alpha(x, y)$ given by

$$\alpha(x, y) = \begin{cases} \min \left\{ \frac{\pi(y) q(x)}{\pi(x) q(y)}, 1 \right\} & \pi(x)q(x) > 0 \\ 1 & \pi(x)q(x) = 0 \end{cases}$$

Thus, actual transitions of the IMH chain take place according to a law P with transition density

$$p(x, y) = q(y)\alpha(x, y), \quad y \neq x$$

and with probability of remaining at the same point given by

$$P(x, \{x\}) = \int q(y)[1 - \alpha(x, y)]\mu(dy)$$

where μ is Lebesgue measure. With this choice of α we have that π is invariant for P .

The perfect IMH algorithm uses the ratios in the acceptance probabilities $\alpha(x, y)$ to reorder the states in such a way that we always accept moves to the left (or downwards). That is, if we write $\pi(x) = kh(x)$ where k is unknown, we define the IMH ordering,

$$x \succeq y \quad \Leftrightarrow \quad \frac{\pi(y)q(x)}{\pi(x)q(y)} \geq 1 \quad \Leftrightarrow \quad \frac{h(y)}{q(y)} \geq \frac{h(x)}{q(x)} \quad (6)$$

With this ordering, we can (hopefully) attain a “lowest state” l . Essentially, one can think of l as the state that is hardest to move away from when running the IMH algorithm. Thus, if we are able to accept a move from l to a candidate state y drawn from the distribution Q with density q , then sample paths from every point in the state space will also accept a move to y , so all possible sample paths will couple. The perfect sampling algorithm is formally described as follows:

IMH (Backward Coupling) Algorithm

1. Draw a sequence of random variables $Q_n \sim Q$ for $n = 0, -1, -2, \dots$, and a sequence $\alpha_n \sim \text{Uniform}(0, 1)$ for $n = -1, -2, \dots$.
2. For each time $-n = -1, -2, \dots$, start a lower path L at l , and an upper path, U at Q_{-n} .
3. (a) For the lower path: Accept a move from l to Q_{-n+1} at time $-n+1$ with probability $\alpha(l, Q_{-n+1})$, otherwise remain at state l . That is, accept the move from l to Q_{-n+1} if $\alpha_{-n} \leq \alpha(l, Q_{-n+1})$.
 (b) For the upper path: Similarly, accept a move from Q_{-n} to Q_{-n+1} at time $-n+1$ if $\alpha_{-n} \leq \alpha(Q_{-n}, Q_{-n+1})$; otherwise remain at state Q_{-n} .
4. Continue until T defined as the first n such that at time $-n+1$ each of these two paths accepts Q_{-n+1} .

By monotonicity, the upper path will accept a candidate point whenever the lower path will, and the two paths will be the same from that time forward. Consequently, our description of the upper process is a formality only, and, indeed, the upper process need not be run at all. Figure 1 illustrates a realization of the perfect IMH algorithm.

For sampling from complicated densities, we may wish to take advantage of the observation that neither the lowest state, l , nor the maximum value $\pi(l)/q(l)$ need be attained explicitly. If we are able to find an m such that

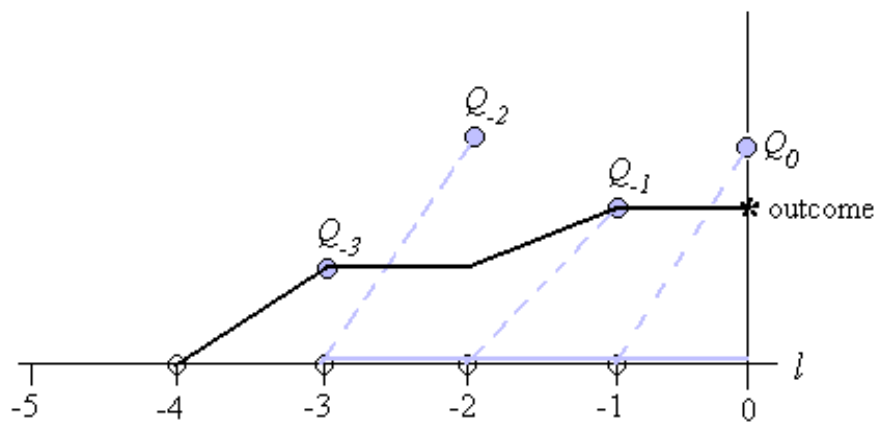
$$m \geq \frac{\pi(x)}{q(x)}, \quad \text{for all } x$$

then we know that

$$\alpha(l, y) \geq \frac{\pi(y)}{m q(y)}$$

so we could modify step 3(a) of the IMH algorithm to read

Figure 1: A Realization of the Perfect IMH Algorithm With Backward Coupling Time at -4



Dashed grey lines represent potential but unrealized arcs of the sample path. Solid grey lines represent the sample paths started at times -1, -2, and -3 that did not achieve the coupling. The solid black line represents the path whose outcome is ultimately observed in the perfect sampling algorithm.

3. (a)' For the lower path: Accept a move from l to Q_{-n+1} at time $-n + 1$ with probability $\alpha(l, Q_{-n+1})$, otherwise remain at state l . That is, accept the move from l to Q_{-n+1} if $\alpha_{-n} \leq \frac{\pi(y)}{m q(y)}$.

We use this modified version of the algorithm to obtain the results in Section 5.

Finally, we refer the reader to Corcoran and Tweedie (2002) for details on how one's choice of Q will affect the expected backward coupling time for the perfect IMH algorithm.

3.2 Slice Coupling

In our context, the state of a sample path at any particular time will be a vector $(\gamma_1, \dots, \gamma_q, \theta_1, \dots, \theta_q, z)$. Therefore, to achieve perfect samples, we will need to be able to couple draws from, for example in the case of the z -component, two different Gamma distributions. We do this via a slice sampling approach which we now describe.

One way to sample from a distribution is to sample uniformly from the region under the plot of its density function. To formalize a method for achieving this, we describe the basis for the *slice sampler* approach.

Suppose we can draw a value x for a random variable X with density function $\pi(x) = c \cdot h(x)$ where the constant of proportionality c is possibly unknown. We then draw a value Y given $X = x$ uniformly over the interval $(0, h(x))$ and finally draw a value X' uniformly from $H(y)$ where $H(y)$ is defined to be the "horizontal slice"

$$H(y) = \{x : h(x) > y\}.$$

It is not difficult to show that X' has density $\pi(x)$. This algorithm is simply a one-iteration stationary version of Gibbs sampling from the density

$$\pi(x, y) \propto I_{\{(x,y): 0 \leq y \leq h(x)\}}$$

where $I_{\{\cdot\}}$ is the indicator function. *Slice sampling* assumes one cannot start with a "draw" from $\pi(x)$; the goal is to get this draw through many Gibbs iterations. *Slice coupling*, on the other hand, assumes that one **can** start with a draw from $\pi(x)$; it is a method we will use to draw potentially common values from two different densities. We now describe this method.

A **naive** strategy for drawing potentially common values from two different densities

$$\pi_1(x) = c_1 h_1(x) \quad \text{and} \quad \pi_2(x) = c_2 h_2(x)$$

is to

1. Draw values x_1 and x_2 from π_1 and π_2 , respectively.
2. Draw values y_1 and y_2 uniformly from the intervals $(0, h_1(x_1))$ and $(0, h_2(x_2))$, respectively.

3. Define the corresponding horizontal slices

$$H_1(y_1) = \{x : h_1(x) > y_1\} \quad \text{and} \quad H_2(y_2) = \{x : h_2(x) > y_2\}.$$

4. If $H_1(y_1) \subseteq H_2(y_2)$ ($H_2(y_2) \subseteq H_1(y_1)$), draw a value uniformly on $H_2(y_2)$ ($H_1(y_1)$).

- If this value is also contained in the other horizontal slice, it is uniform over that slice as well, and is therefore a common value drawn from π_1 and π_2 .
- If this value is not contained in the other horizontal slice, we have a resulting value from one distribution. Draw the resulting value from the other distribution uniformly from the other horizontal slice.

In step 4 the assumption that one horizontal slice is contained in the other is taken merely for illustration purposes and can be removed when applying different techniques on drawing the uniform random number in that step. As written, the algorithm may not have a high success rate for achieving common draws. Furthermore, for purposes of coupling sample paths of Markov chains, one must take care to run each path in the same way. This includes using the same random numbers to draw both x_1 and x_2 in step 1, using a single uniform(0,1) random number to produce y_1 and y_2 in step 2, and drawing the final value in part 2 of step 4 in a “meaningful” way (for example, with a series of accept/reject steps).

We refer the reader to Wilson (2000) and Corcoran and Schneider (2001) for further information on this technique including shifting and scaling methods for increasing the likelihood of a successful coupling between distributions with different locations and shapes.

4 Perfect Simulation from the Posterior

In Section 4.1 we describe an existing algorithm of Huang and Djurić (2002) for perfect sampling from (2) when σ^2 and $\boldsymbol{\theta}$ are fixed and known. In the remaining sections we gradually allow for other layers of randomness.

4.1 Fixed Variance, Fixed Coefficients

For fixed σ^2 and $\boldsymbol{\theta}$, the posterior distribution of the indicators in (1) with the uniform noninformative prior on the components of $\boldsymbol{\gamma}$ is

$$\pi_{\boldsymbol{\Gamma}|\mathbf{Y}}(\boldsymbol{\gamma}|\mathbf{y}) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^q \sum_{k=1}^q \gamma_j \gamma_k \theta_j \theta_k \mathbf{x}'_j \mathbf{x}_k + \frac{1}{\sigma^2} \sum_{j=1}^q \gamma_j \theta_j \mathbf{x}'_j \mathbf{y}\right). \quad (7)$$

The full conditional distributions are then given by

$$\begin{aligned} \pi(\gamma_i = 1 | \boldsymbol{\gamma}_{-i}, \mathbf{y}) &= \frac{Pr(\gamma_i = 1, \boldsymbol{\gamma}_{-i} | \mathbf{y})}{Pr(\gamma_i = 0, \boldsymbol{\gamma}_{-i} | \mathbf{y}) + Pr(\gamma_i = 1, \boldsymbol{\gamma}_{-i} | \mathbf{y})} \\ &= \left[1 + \exp\left(\frac{1}{\sigma^2} \sum_{\substack{j=1 \\ j \neq i}}^q \gamma_j \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j + \frac{1}{2\sigma^2} \theta_i^2 \mathbf{x}'_i \mathbf{x}_i - \frac{1}{\sigma^2} \theta_i \mathbf{x}'_i \mathbf{y}\right) \right]^{-1} \end{aligned} \quad (8)$$

where $\gamma_{-i} = (\gamma_1, \dots, \gamma_{i-1}, \gamma_{i+1}, \dots, \gamma_q)$.

A first-pass perfect sampling approach would be to consider the partial ordering

$$\Gamma_1 \preceq \Gamma_2$$

when any component that is a 1 in Γ_1 is also a 1 in Γ_2 . One could then consider the “top” and “bottom” of the space to be $(1, \dots, 1)$ and $(0, \dots, 0)$, respectively. Unfortunately, Huang and Djurić (2002) show that the standard Gibbs update does not, in general, preserve the monotonicity property required for the feasibility of a perfect sampling scheme for large q . Instead they make clever use of the Gibbs coupler which is a *support set coupling* technique that we now describe.

To begin, we require lower and upper bounding processes for the Gibbs update given by (8). That is, we seek, for $n = 1, 2, \dots$, processes $\{L_{-n}\}$ and $\{U_{-n}\}$ so that

$$L_{-n} \preceq X_{-n} \preceq U_{-n},$$

where $\{X_{-n}\}$ is the q -dimensional Gibbs chain with stationary distribution given by (7) and updates given by (8).

Huang and Djurić (2002) construct $\{L_{-n}\}$ and $\{U_{-n}\}$ by assigning, for any time step, the i th components to be 1 with probabilities $P_L(\gamma_i = 1)$ and $P_U(\gamma_i = 1)$, respectively, where these probabilities are such that

$$P_L(\gamma_i = 1) \leq \pi(\gamma_i = 1 | \gamma_{-i}, \mathbf{y}) \leq P_U(\gamma_i = 1).$$

To obtain such *sandwich distributions*, we begin by breaking the sum in (8) into two parts

$$\pi(\gamma_i = 1 | \gamma_{-i}, \mathbf{y}) = \left[1 + \exp \left\{ \frac{1}{\sigma^2} \left(\sum_{j \in C} \gamma_j \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j + \sum_{j \notin C} \gamma_j \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j + \frac{1}{2} \theta_i^2 \mathbf{x}'_i \mathbf{x}_j - \theta_i \mathbf{x}'_i \mathbf{y} \right) \right\} \right]^{-1} \quad (9)$$

where C is the set of indices of all of the components that have coupled successfully by this particular time step. (We have suppressed the time notation.)

For the γ -components referenced by C , all three processes are equal, and for the γ -components not in C , the current state of these components are still unknown. Using the second sum in (9) we can make the overall expression smallest if we set γ_j with $j \notin C$ to be 1 when the coefficient $\theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j$ is positive and, likewise, we can make it largest if we set γ_j with $j \notin C$ to be 1 when the coefficient $\theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j$ is negative. The other γ -components are set to zero.

So, we have

$$P_L(\gamma_i = 1) = \left[1 + \exp \left\{ \frac{1}{\sigma^2} \left(\sum_{j \in C} \gamma_j \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j + \sum_{j \in P} \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j + \frac{1}{2} \theta_i^2 \mathbf{x}'_i \mathbf{x}_i - \theta_i \mathbf{x}'_i \mathbf{y} \right) \right\} \right]^{-1} \quad (10)$$

$$(11)$$

and

$$P_U(\gamma_i = 1) = \left[1 + \exp \left\{ \frac{1}{\sigma^2} \left(\sum_{j \in C} \gamma_j \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j + \sum_{j \in N} \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j + \frac{1}{2} \theta_i^2 \mathbf{x}'_i \mathbf{x}_i - \theta_i \mathbf{x}'_i \mathbf{y} \right) \right\} \right]^{-1} \quad (12)$$

$$(13)$$

where

$$P = \{j \mid \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j > 0\} \quad \text{and} \quad N = \{j \mid \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j < 0\}.$$

Rather than following the values of the γ -components through time, support set coupling keeps track of the *potential* values generated by the Gibbs sampler. For each $i = 1, 2, \dots, q$, we will determine, at each time step, if γ_i is 0, 1, or undecided. That is, we will assign its support set, S_i , to be $\{0\}$, $\{1\}$, or $\{0, 1\}$. It is most difficult to obtain, for the value of γ_i , a 1 in the lower process and a 0 in the upper process. Hence, if the former is achieved, all possible sample paths will have γ_i set equal to 1 ($S_i = \{1\}$). In the latter case, all possible sample paths will have γ_i set equal to zero ($S_i = \{0\}$). If neither case arises, we will leave the value of γ_i in our sample path of interest undecided ($S_i = \{0, 1\}$). More specifically, at each backward time step $n = 1, 2, \dots$ and for each component $i = 1, 2, \dots, q$, we draw a uniform(0,1) random variable V , ($V = V_i^{-n}$), and set

$$S_i = \begin{cases} \{0\}, & \text{if } V > P_U(\gamma_i = 1) \\ \{1\}, & \text{if } V < P_L(\gamma_i = 1) \\ \{0, 1\}, & \text{otherwise} \end{cases}$$

Coupling is achieved when all q components have a singleton support set.

We wish to strongly emphasize here that we have suppressed important notation in an attempt to not (we hope) overwhelm the reader in order to get some general ideas across. In particular, we have suppressed

1. General Time Notation

The organization of random number streams, that is, the “re-use” of random numbers, is critical to the success of any perfect simulation algorithm. Newcomers to perfect simulation algorithms might best be served by first gaining an understanding of the basic ideas involved. We again refer interested readers to Casella et al. (2000).

2. Gibbs Component Time Notation:

In the update probabilities given by (10) and (12), we are revising the q γ -components in order at each time step. If we use $\gamma_i^{(n)}$ to represent the value of the i th component at time n , the update for $\gamma_i^{(n)}$ involves $\gamma_j = \gamma_j^{(n)}$ for $j = 1, \dots, i - 1$ and $\gamma_j = \gamma_j^{(n-1)}$ for $j = i + 1, \dots, q$.

Expanded notation may be found in Huang and Djurić (2002).

4.2 Random Variance, Fixed Coefficients

We first extend the model by incorporating a random variance for the Gaussian noise according to (3). Assuming that the coefficient-vector $\boldsymbol{\theta}$ is fixed, the posterior distribution becomes

$$\pi_{\boldsymbol{\Gamma}, Z | \mathbf{Y}}(\boldsymbol{\gamma}, z | \mathbf{y}) \propto z^{\frac{n}{2} + \frac{\nu}{2} - 1} \exp\left\{-\frac{1}{2}z\left[\frac{\lambda\nu}{2} + (\mathbf{y} - \sum_{i=1}^q \gamma_i \boldsymbol{\theta}_i \mathbf{x}_i)'(\mathbf{y} - \sum_{i=1}^q \gamma_i \boldsymbol{\theta}_i \mathbf{x}_i)\right]\right\} \quad (14)$$

Since we intend to apply a Gibbs-sampler again, we need to look at the conditional probabilities. It is easy to see that

$$Z | \boldsymbol{\Gamma}, \mathbf{Y} \sim \Gamma\left(\frac{\lambda + \nu}{2}, \frac{1}{2}\left[(\mathbf{y} - \sum_{i=1}^q \gamma_i \boldsymbol{\theta}_i \mathbf{x}_i)'(\mathbf{y} - \sum_{i=1}^q \gamma_i \boldsymbol{\theta}_i \mathbf{x}_i) + \lambda\nu\right]\right) \quad (15)$$

and that the posterior conditional distribution for the vector of indicators $\boldsymbol{\gamma}$ given Z is

$$\pi_{\boldsymbol{\Gamma} | \mathbf{Y}, Z}(\boldsymbol{\gamma} | \mathbf{y}, z) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^q \sum_{k=1}^q \gamma_j \gamma_k \boldsymbol{\theta}_j \boldsymbol{\theta}_k \mathbf{x}_j' \mathbf{x}_k + \frac{1}{\sigma^2} \sum_{j=1}^q \gamma_j \boldsymbol{\theta}_j \mathbf{x}_j' \mathbf{y}\right) \quad (16)$$

where a random variable Y has a Gamma distribution $\Gamma(\alpha, \beta)$ if it has density function $f_Y(y) = \frac{\beta}{\Gamma(\alpha)} (\beta y)^{\alpha-1} e^{-\beta y} I_{(0, \infty)}(y)$.

Note that the distribution for $\boldsymbol{\Gamma} | \mathbf{Y}, Z$ is the same as in (7) since $Z = z$ is fixed. So once we know the value for Z , we can use the support set coupling from Section 4.1 to simulate perfectly from $\boldsymbol{\Gamma} | \mathbf{Y}, Z$. On the other hand, if we know the value for $\boldsymbol{\Gamma}$, we can draw from a Gamma distribution for $Z | \boldsymbol{\Gamma}, \mathbf{Y}$. Hence, we can run sample paths of a “bivariate” Markov chain that has stationary distribution given by (14). (Note that we are thinking of $\boldsymbol{\gamma}$ and Z as two random variables as opposed to $q + 1$ random variables as $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_q)'$ since the algorithm of Section 4.1 returns a vector $\boldsymbol{\gamma}$.)

Our problem now is to consider coupling together these Gibbs-generated sample paths. Note that we will be describing a perfect simulation within a perfect simulation.

The Coupling

The key observation for simulating from (14) is that the conditional Gamma distribution for $Z | \boldsymbol{\Gamma}, \mathbf{Y}$ has a **fixed** shape parameter $\alpha = \frac{n+\nu}{2}$ and scale parameter $\beta(\boldsymbol{\gamma}) = \frac{1}{2}\left[(\mathbf{y} - \sum_{i=1}^q \gamma_i \boldsymbol{\theta}_i \mathbf{x}_i)'(\mathbf{y} - \sum_{i=1}^q \gamma_i \boldsymbol{\theta}_i \mathbf{x}_i) + \lambda\nu\right]$ that can be bounded above and below *independently* of $\boldsymbol{\gamma}$ (see below for details):

$$\beta_{min} \leq \beta(\boldsymbol{\gamma}) \leq \beta_{max} \quad \text{for all } \boldsymbol{\gamma} \in \{0, 1\}^q. \quad (17)$$

If we can get a common draw for $\Gamma(\alpha, \beta_{min})$ and $\Gamma(\alpha, \beta_{max})$, it will be a draw from $\Gamma(\alpha, \beta)$ for all β such that $\beta_{min} \leq \beta \leq \beta_{max}$, and therefore a draw for $Z|\Gamma, \mathbf{Y}$ no matter what the particular values for $(\gamma_1, \dots, \gamma_q)$ were! This can be achieved by the slice-coupling approach described in Section 3.2 and we address this in more detail in Section 4.2.1. So, at each time point $-n$, $n = 1, 2, \dots$, we begin by drawing two values for Z , one from each of the distributions $\Gamma(\alpha, \beta_{min})$ and one from $\Gamma(\alpha, \beta_{max})$ using the slice coupling approach described in Section 3.2 until the first time point $-n^*$ where we get a common value for Z . Once this value is locked in, all ambiguity is removed and we move forward in time from $-n^*$ to zero with a Gibbs sampler where we alternate at each time point between

1. using Huang and Djurić's algorithm (Section 4.1) in it's entirety to sample a vector of γ components given Z
2. sampling a new Z from the resulting gamma distribution described by (15) for the fixed vector γ from step 1

Bounding β

Expanding the expression for $\beta(\gamma)$, we get

$$\beta(\gamma) = \frac{1}{2}(\mathbf{y}'\mathbf{y} + \lambda\nu) - \sum_{i=1}^q \gamma_i \theta_i \mathbf{x}'_i \mathbf{y} + \frac{1}{2} \sum_{i=1}^q \gamma_i^2 \theta_i^2 \mathbf{x}'_i \mathbf{x}_i + \sum_{i < j} \gamma_i \gamma_j \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j$$

By defining

$$\begin{aligned} I_1 &= \{ i \mid \theta_i \mathbf{x}'_i \mathbf{y} \geq 0 \} \\ I_2 &= \{ i \mid \theta_i \mathbf{x}'_i \mathbf{y} < 0 \} \\ M_1 &= \{ (i, j) \mid \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j \geq 0 \} \\ M_2 &= \{ (i, j) \mid \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j < 0 \} \end{aligned}$$

and

$$\begin{aligned} \beta_{min} &:= \max \left\{ \frac{1}{2} \lambda \nu, \frac{1}{2} (\mathbf{y}'\mathbf{y} + \lambda \nu) - \sum_{i \in I_1} \theta_i \mathbf{x}'_i \mathbf{y} + \sum_{(i,j) \in M_2} \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j \right\} \\ \beta_{max} &:= \frac{1}{2} (\mathbf{y}'\mathbf{y} + \lambda \nu) - \sum_{i \in I_2} \theta_i \mathbf{x}'_i \mathbf{y} + \frac{1}{2} \sum_{i=1}^q \gamma_i^2 \theta_i^2 \mathbf{x}'_i \mathbf{x}_i + \sum_{(i,j) \in M_1} \theta_i \theta_j \mathbf{x}'_i \mathbf{x}_j \end{aligned}$$

(17) is easily verified.

4.2.1 Slice Coupling for the z -component

We now demonstrate how to couple the z -component using the slice sampling approach from Section 3.2 in more detail.

Our goal is to get a potentially common draw from two different Gamma distributions, $\Gamma(\alpha, \beta_{min})$ and $\Gamma(\alpha, \beta_{max})$. Section 3.2 shows that we only need to know the density function up to a constant, so we choose $h_1(x) = (\beta_{min}x)^{\alpha-1}e^{-\beta_{min}x}$, and $h_2(x) = (\beta_{max}x)^{\alpha-1}e^{-\beta_{max}x}$, respectively.

We assume that $\alpha \leq 1$ (the case where $\alpha > 1$ works in a similar manner) and we clearly have $\beta_{min} < \beta_{max}$.

Note that the horizontal slice $H(y)$ in the last step if the slice sampling procedure is of the form $(0, r)$. Also note that a *smaller* scale parameter β will yield a *larger* Gamma variable and a *wider* horizontal slice $H(y)$.

1. Draw $x \sim \Gamma(\alpha, 1)$, let $x_1 = \frac{x}{\beta_{min}}$, and $x_2 = \frac{x}{\beta_{max}}$.
2. Draw $u \sim \text{Uniform}(0, 1)$ and let $y_1 = uh_1(x_1)$, and $y_2 = y_1$.
3. Approximate the endpoint with r_1 of the wider slice and
 - (a) draw $u' \sim \text{Uniform}(0, r_1)$
 - (b) if $h_1(u') > y_1$, accept $x'_1 = u'$ otherwise update $r_1 = u'$ and go to step 1.
4. If $h_2(x'_1) > y_2$, accept $x'_2 = x'_1$ - a common draw!
otherwise (accept-reject): approximate the endpoint $r_2 = x'_1$
 - (a) draw $u' \sim \text{Uniform}(0, r_2)$
 - (b) if $h_2(u') > y_2$, accept $x'_2 = u'$
otherwise update $r_2 = u'$ and go to step 1.

Remarks

- Note that our choice of h_1 and h_2 yields $h_1(x_1) = h_2(x_2)$ (with the notation from step 1), which justifies setting $y_2 = y_1$ in step 2.
- Even though the pdf of a Gamma distribution is non-invertible, this did not create a problem in steps 3 and 4 of the slice sampling procedure. We may be approximating the right endpoint of the horizontal slice, but we are drawing uniformly from the true horizontal slice in a accept/reject procedure by checking whether the draw from the approximated slice falls under the density or not.

4.2.2 Sampling for the variance and indicators together

An algorithm to draw from (14) is given by:

1. Find a *backward coupling time* T : for each time $t = 0, -1, -2, \dots$
 - (a) Draw z_t^{min} and z_t^{max} according to $\Gamma(\alpha, \beta_{min})$ and $\Gamma(\alpha, \beta_{max})$ using the slice sampling procedure described in Section 4.2.1.
 - (b) If $z_t^{min} = z_t^{max}$, set $T \leftarrow t$ and go to step 2 (*coalescence*).
 - (c) Otherwise, set $t \leftarrow t - 1$ and go to step 1a.
2. Draw $\gamma_T \sim \mathbf{\Gamma} | (\mathbf{Y}, Z = z_T)$ using the support set coupling from Section 4.1.
3. For each time $t = T + 1, \dots, -1, 0$ (*outer Gibbs sampler*):
 - (a) Draw $z_t \sim Z | (\mathbf{Y}, \mathbf{\Gamma} = \gamma_{t-1})$ using the slice sampling procedure from Section 4.2.1.
 - (b) Draw $\gamma_t \sim \mathbf{\Gamma} | (\mathbf{Y}, Z = z_t)$ using the support set coupling from Section 4.1 (*inner Gibbs sampler*).
4. (γ_0, z_0) is a draw from the posterior (19).

4.3 Fixed Variance, Random Coefficients

We now go back to assuming that the variance σ^2 is fixed, but we allow the coefficient-vector θ to be random according to (4).

Combining γ and θ

We combine the random components γ_i and θ_i by defining $\beta_i := \gamma_i \theta_i$ ($i = 1, \dots, q$) to have the appropriate mixture distribution. Ultimately, since we are maximizing a marginal distribution, we are only interested in the values for γ which can be “recovered” from β by setting

$$\gamma_i = \begin{cases} 0 & \text{if } \beta_i = 0 \\ 1 & \text{if } \beta_i \neq 0 \end{cases} \quad i = 1, \dots, q.$$

Let $g_{\mathbf{B}}(\beta)$ denote the pdf for $\beta = (\beta_1, \dots, \beta_q)'$ and let $L(\beta) = \exp\{-\frac{1}{2\sigma^2}(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)'(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)\}$ be the likelihood. Then the posterior distribution we want to draw from is

$$\pi_{\beta | \mathbf{Y}}(\beta | \mathbf{y}) \propto L(\beta) \times g_{\mathbf{B}}(\beta) \tag{18}$$

Using IMH

To simulate from the posterior (18), we apply the IMH-algorithm from Section 3.1. We use the prior distribution for $\boldsymbol{\beta}$ as the candidate distribution and set $q(\boldsymbol{\beta}) = g_{\mathbf{B}}(\boldsymbol{\beta})$. With this choice we can easily simulate candidates by drawing $\boldsymbol{\beta} \sim N(\boldsymbol{\xi}, \sigma^2 V)$ and setting each component β_i equal to 0 with probability $\frac{1}{2}$.

For the IMH algorithm, we also need to know that

$$\max_{\boldsymbol{\beta}} \frac{\pi}{q} = \max_{\boldsymbol{\beta}} \frac{L(\boldsymbol{\beta})g_{\mathbf{B}}(\boldsymbol{\beta})}{g_{\mathbf{B}}(\boldsymbol{\beta})} = \max_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) = \max_{\boldsymbol{\beta}} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)'(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)\right\} \leq 1.$$

So an algorithm to simulate from (18) can be written as:

1. Find a *backward coupling time* T : for each time $t = 0, -1, -2, \dots$
 - (a) Draw $\boldsymbol{\beta}_t \sim q(\cdot)$ and $u_t \sim \text{Uniform}(0, 1)$.
 - (b) If $u_t \leq L(\boldsymbol{\beta}_t) \times 1$, set $T \leftarrow t$ and go to step 2 (*coalescence*).
 - (c) Otherwise, set $t \leftarrow t - 1$ and go to step 1a.
2. Set $X_T = q_T$ and for each time $t = T + 1, T + 2, \dots, 0$:
 - (a) If $u_t \leq \frac{L(X_{t-1})}{L(\boldsymbol{\beta}_t)}$ (*accept the candidate*), set $X_t = \boldsymbol{\beta}_t$.
 - (b) Otherwise (*reject the candidate*), set $X_t = X_{t-1}$.
 - (c) Set $t \leftarrow t + 1$.
3. X_0 is a draw from the posterior (18)!

4.4 Random variance, random coefficients

In this section, we incorporate both a random variance and random coefficients with prior distributions according to 3 and 4. We solve a special case assuming that the coefficients β_i are uncorrelated and that q is “small”.

4.4.1 Uncorrelated coefficients and small q

We again combine γ_i and θ_i into one variable $\beta_i = \gamma_i \theta_i$. To find the posterior, we assume that the coefficients β_i ($i = 1, \dots, q$) are uncorrelated, i.e. we let the prior distribution for $\boldsymbol{\beta}$ be $N(\boldsymbol{\xi}, \sigma^2 \mathbf{I})$. Note that we also have the random component $Z = \frac{1}{\sigma^2}$. The density for the (joint) prior distribution is

$$g_{\mathbf{B}, Z}(\boldsymbol{\beta}, z) = g_{\mathbf{B}|Z}(\boldsymbol{\beta}|z)g_Z(z) \propto \prod_{i=1}^q \left(\delta_0(\beta_i) + \mathbb{1}_{\{\beta \neq 0\}} \sqrt{\frac{z}{2\pi}} \exp\left\{-\frac{z}{2}(\beta_i - \mu_i)^2\right\} \right) \times z^{\frac{q}{2}-1} \exp\left\{-\frac{\lambda\nu}{2}z\right\}$$

where $\delta_0(\cdot)$ is the Dirac-delta function that satisfies $\delta_0(\beta) = 0$ if $\beta \neq 0$ and $\int_A \delta(\beta) d\beta = 1$ for every set A with $0 \in A$.

The likelihood is given by:

$$L(\boldsymbol{\beta}, z) = z^{\frac{n}{2}} \exp\left\{-\frac{z}{2}(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)'(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)\right\}.$$

With this notation, the distribution we wish to draw from is given by

$$\pi_{\mathbf{B}, Z | \mathbf{Y}}(\boldsymbol{\beta}, z | \mathbf{y}) \propto L(\boldsymbol{\beta}, z) \times g_{\mathbf{B}, Z}(\boldsymbol{\beta}, z). \quad (19)$$

Using IMH

We again apply the IMH-algorithm from Section 3.1 and set $q(\boldsymbol{\beta}, z) = z^{\frac{n}{2}} g_{\mathbf{B}, Z}(\boldsymbol{\beta}, z)$ as the candidate density. With this choice of $q(\boldsymbol{\beta}, z)$, we get that

$$\max \frac{\pi}{q} = \max_{(\boldsymbol{\beta}, z)} \frac{L(\boldsymbol{\beta}, z)}{z^{\frac{n}{2}}} = \max_{(\boldsymbol{\beta}, z)} \exp\left\{-\frac{z}{2}(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)'(\mathbf{y} - \sum_{i=1}^q \beta_i \mathbf{x}_i)\right\} \leq 1.$$

A Gibbs-sampler for the candidates

The simulation of the candidates is less straightforward in this case we employ a Gibbs-sampler to draw from $q(\boldsymbol{\beta}, z)$. It can be shown that the conditional probabilities are given by:

$$\begin{aligned} \mathbf{B} | Z & : \beta_i | Z = \begin{cases} 0 & \text{w.p. } \frac{1}{2} \\ \sim N(\xi_i, \frac{1}{z}) & \text{w.p. } \frac{1}{2} \end{cases} \quad i = 1, \dots, q \\ Z | \mathbf{B} & \sim \Gamma(a(\boldsymbol{\beta}), b(\boldsymbol{\beta})) \end{aligned}$$

$$a(\boldsymbol{\beta}) = \frac{1}{2}[n + \nu + \sum_{i=1}^q \mathbb{1}_{\{\beta_i \neq 0\}}] \text{ and } b(\boldsymbol{\beta}) = \frac{1}{2}[\lambda\nu + \sum_{i=1}^q \mathbb{1}_{\{\beta_i \neq 0\}}(\beta_i - \xi_i)^2].$$

Again, it is easy to draw from the “two” ($\boldsymbol{\beta}$ and z) components for the Gibbs-sampler once the other component is known. To get coalescence, we make the following observation: Both the shape-parameter α and the scale parameter β depend on the current values of the $\boldsymbol{\beta}$ -vector. If, however, all components β_i are equal to zero (an event with positive probability 2^{-q}), we know that the shape and scale parameter are $\alpha(\mathbf{0}) = \frac{1}{2}(n + \nu)$ and $\beta(\mathbf{0}) = \frac{1}{2}\lambda\nu$. That means that we need to sample from $\mathbf{B} = \boldsymbol{\beta}$ (going back in time) until all components of $\boldsymbol{\beta}$ are zero. Then we can draw z from $Z \sim \Gamma(\alpha(\mathbf{0}), \beta(\mathbf{0}))$ to initialize a Gibbs-sampler and alternately sample from the conditional distributions forward to time $t = 0$.

So, an algorithm to simulate candidates from $q(\boldsymbol{\beta}, z)$ is given can be written as:

1. Find a *backward coupling time* T : for each time $t = 0, -1, -2, \dots$
 - (a) For each $i = 1, \dots, q$ draw $u_i^{(t)}$ and if $u_i^{(t)} < \frac{1}{2}$, set $\beta_i^{(t)} = 0$.
 - (b) If $\beta_i^{(t)} = 0$ for all $i = 1, \dots, q$, go to step 2 (*coalescence*)
 - (c) Otherwise, set $t \leftarrow t - 1$ and go to 1a.
2. Set $\alpha = \alpha(\mathbf{0})$, $\beta = \beta(\mathbf{0})$ and draw $z_T \sim \Gamma(\alpha, \beta)$
3. For each time $t = T + 1, \dots, -1, 0$:
 - (a) Draw $\beta_t \sim N(\xi, \frac{1}{z_{t-1}} \mathbf{I})$. For each $i = 1, \dots, q$, draw $u_i^{(t)}$ and if $u_i^{(t)} < \frac{1}{2}$, set $\beta_i^{(t)} = 0$.
 - (b) Set $\alpha = \alpha(\beta_t)$, $\beta = \beta(\beta_t)$ and draw $z_t \sim \Gamma(\alpha, \beta)$.
 - (c) Set $t \leftarrow t + 1$.

Remarks

- The algorithm above describes how to get samples from the candidate distribution $q(\beta, z)$. To get draws from the posterior (19), this procedure has to be incorporated into an IMH-algorithm.
- The expected backward coupling time when Gibbs-sampling for the candidates is $E[T] = 2^q$ which makes this algorithm suitable for very small values of q only.

There is still the open question as to how one might develop an algorithm to sample from the posterior in the case of correlated coefficients.

5 Simulation Results

To test the performance of the algorithms, we used simulated data to mimic the scenario of the different models. In all cases, the predictors \mathbf{x}_i were generated independent and identically distributed according to $\mathbf{x}_i \sim N(\mathbf{0}, \mathbf{I})$, $i = 1, \dots, q$.

For each different model, we describe in detail how the test data was produced. Results are presented by listing the frequencies for the γ -vectors as well as specifying the marginal probabilities for each component γ_i . To discuss computational cost we list the average, the minimum, and the maximum backward coupling time.

5.1 Fixed Variance, Fixed Coefficients

This case has been solved and tested in Huang and Djurić (2002). We include our test results for completeness.

We used $q = 5$ predictors and generated $n = 20$ data records the following way: After simulating a noise vector $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$ of size 20 (σ^2 was set to 1), the data was computed by assigning $\mathbf{y} = \sum_{i=1}^5 \gamma_i \theta_i \mathbf{x}_i + \boldsymbol{\varepsilon}$ with $\boldsymbol{\theta} = (0.8, 0.7, 0.7, 0.7, 0.9)'$ and $\boldsymbol{\gamma} = (1, 0, 0, 1, 0)'$.

After 1000 iterations, we obtained results shown in Table 1.

The average *backward coupling time* for the Gibbs-sampler was $T = 1.686$ with a minimum of $T = 1$ and a maximum of $T = 3$.

The results show that the algorithm could easily recover which of the predictors had been part of the model. Both lists clearly reveal the $\boldsymbol{\gamma}$ -vector that has been used to generate the data \mathbf{y} .

$\boldsymbol{\gamma}$	percentage	component	percentage
(1,0,0,1,0)	92.9 %	$P(\gamma_1 = 1)$	99.3 %
(1,1,0,1,0)	4.7 %	$P(\gamma_2 = 1)$	4.9 %
(1,0,0,1,1)	1.2 %	$P(\gamma_3 = 1)$	0.4 %
(0,0,0,1,0)	0.6 %	$P(\gamma_4 = 1)$	99.9 %
(1,0,1,1,0)	0.3 %	$P(\gamma_5 = 1)$	1.3 %
(0,1,0,1,0)	0.1 %		
(1,1,1,1,0)	0.1 %		
(1,0,0,0,1)	0.1 %		

Table 1: Results after 1000 iterations (with fixed variance and fixed coefficients).

5.2 Random Variance, Fixed Coefficients

To test this case, we simulated a value z according to $Z \sim \Gamma(\frac{\nu}{2}, \frac{\lambda\nu}{2})$ and used $\sigma^2 = \frac{1}{z}$ as the variance for the noise vector $\boldsymbol{\varepsilon}$. We again used $q = 5$ predictors, but this time generated $n = 50$ data records. The coefficient vector of the model was $\boldsymbol{\theta} = (1.2, 1.3, 1, 1.1, 1.2)'$ and the hyperparameters were chosen to be $\lambda = 1$ and $\nu = 1$. The predictors that went into the model were determined by $\boldsymbol{\gamma} = (1, 0, 0, 1, 0)'$.

After 1000 iterations, we obtained the results shown in Table 2 (we only list realizations of $\boldsymbol{\gamma}$ that had a frequency of 1% or more).

The average backward coupling time for the coupling the z -component (“outer Gibbs sampler”) was $T = 30.091$ with a minimum of $T = 1$ and a maximum of $T = 366$.

For the “inner Gibbs sampler” – i.e. for the support set coupling to draw the $\boldsymbol{\gamma}$ -component – the average backward coupling time was $T = 1.141$ with a minimum of $T = 1$ and a maximum of $T = 3$.

In this case, the algorithm also detected the right predictors, with the second most frequent model only occurring half as often as the true one. Again, the marginal probabilities for each component γ_i clearly point to the correct $\boldsymbol{\gamma}$ -vector.

γ	percentage	component	percentage
(1,0,0,1,0)	37.8 %	$P(\gamma_1 = 1)$	80.9 %
(1,0,1,1,0)	18.2 %	$P(\gamma_2 = 1)$	0.6 %
(0,0,0,1,0)	11 %	$P(\gamma_3 = 1)$	0.28 %
(1,0,0,1,1)	8.9 %	$P(\gamma_4 = 1)$	88.9 %
(1,0,0,0,0)	5.1 %	$P(\gamma_5 = 1)$	15.7.3 %
(0,0,1,1,0)	3.5 %		
(1,1,0,1,0)	3.1 %		
(1,0,1,1,1)	2.3 %		
(1,0,1,0,0)	1.6 %		
(1,0,0,0,1)	1.3 %		
(0,0,0,1,1)	1 %		

Table 2: Results after 1000 iterations (with random variance and fixed coefficients).

5.3 Fixed Variance, Random Coefficients

This case was tested with $q = 5$ predictors and $n = 20$ data records. We generated 20 coefficient vectors θ according to $\Theta \sim N(\xi, \sigma^2 \mathbf{I})$ with $\xi = (1, 1, 1, 1, 1)'$ and $\sigma^2 = 0.5$ which was also used as the variance for the noise vector ε . The predictors that went into the model were determined by $\gamma = (1, 0, 0, 1, 0)'$.

After 100 iterations, we obtained the results shown in Table 3.

The average *backward coupling time* for the IMH-algorithm was $T = 429842.21$ with a minimum of $T = 8809$ and a maximum of $T = 2601550$.

Note that the high backward coupling times in this model do not allow to go to much higher dimensions. It is possible that this drawback may be overcome with the idea of a *multistage coupler* which is a subject of future work. See the Appendix on page 19 for more details on the multistage coupler.

Despite the small number of iterations, the algorithm clearly yields the correct model.

γ	percentage	component	percentage
(1,0,0,1,0)	59 %	$P(\gamma_1 = 1)$	100 %
(1,0,0,1,1)	28 %	$P(\gamma_2 = 1)$	0.07 %
(1,1,0,1,0)	6 %	$P(\gamma_3 = 1)$	0.06 %
(1,0,1,1,0)	4 %	$P(\gamma_4 = 1)$	100 %
(1,0,1,1,1)	2 %	$P(\gamma_5 = 1)$	31 %
(1,1,0,1,1)	1 %		

Table 3: Results after 100 iterations (with fixed variance and random coefficients).

5.4 Random Variance, Random Coefficients

5.4.1 I.I.D. Coefficients for Small q

We used $q = 3$ predictors and $n = 20$ data records. The mean vector for the coefficients was given by $\xi = (1, 1, 1)'$ and the hyperparameters were chosen to be $\lambda = 10$ and $\nu = 1$. The predictors that went into the model were determined by $\gamma = (1, 1, 0)'$.

After 100 iterations, we obtained the results shown in Table 4.

The average backward coupling time for the IMH-algorithm was $T = 542.8$ with a minimum of $T = 2$ and a maximum of $T = 4257$. The average backward coupling time for the Gibbs-sampler for drawing the candidates was $T = 7.53$.

Due to the very large backward coupling times of the IMH algorithm in higher dimensions, we chose only 3 predictors and 20 data records in this case. Again, the *multistage coupler* described in the Appendix on page 19 might help to reduce these coupling times. This would allow higher dimensions in n , but the expected backward coupling time for the Gibbs-sampler will still be 2^q .

Again, the algorithm clearly detected the correct γ -vector.

γ	percentage
(1,1,0)	34 %
(1,0,0)	18 %
(1,1,1)	18 %
(0,1,0)	13 %
(0,1,1)	9 %
(0,0,1)	3 %
(1,0,1)	3 %
(0,0,0)	2 %

component	percentage
$P(\gamma_1 = 1)$	73 %
$P(\gamma_2 = 1)$	74 %
$P(\gamma_3 = 1)$	33 %

Table 4: Results after 100 iterations (with random variance and random coefficients).

Appendix – Towards a Multistage Coupler

The following idea could help to drastically reduce the large backward coupling times for the IMH algorithm:

The state space is partitioned into two clusters. In a first step, a procedure determines into which two cluster the current draw will fall. Then, in second step, a draw from the target distribution is made *restricted* to that particular cluster. (This general idea is due to Meng (2000) and also mentioned in Murdoch (2000).)

To describe this idea in more detail within the framework of model selection, assume that we would like to draw from a target density $\pi(x) \propto L(x)q(x)$ and that intend to apply an IMH-algorithm with candidate density $q(x)$. Moreover, suppose that $\max_x \frac{\pi(x)}{q(x)} = \max_x L(x) \leq 1$.

First, the partition of the state space S is made the following way: We define a *high likelihood cluster* $M_1 := \{x \in X | L(x) \geq k\}$ and a *low likelihood cluster* $M_2 := X \setminus M_1 = \{x \in X | L(x) < k\}$ for a fixed k that is small “enough” (how k should be chosen is explained later on).

In the first stage we perform an “IMH-like” step, but we only go back in time far enough to determine in which *cluster* the draw X_0 falls into:

First stage

1. Draw a candidate $Q \sim q(\cdot)$ and a $U \sim \text{Uniform}(0, 1)$.
2. If $U < \frac{L(Q)}{k}$ (we then have $U < \frac{L(Q)}{k} < \frac{L(Q)}{L(y)}$ for all $y \in M_2$, so that every element in M_2 will *accept*):
 - (a) If $Q \in M_1$, then $X_0 \in M_1$: *report M_1 as the cluster*
 - (b) If $Q \in M_2$, do a full IMH-step and *report the cluster of X_0*
3. If $U \geq \frac{L(Q)}{k}$ (we then have $U \geq \frac{L(Q)}{k} \geq \frac{L(Q)}{L(y)}$ for all $y \in M_1$, so that every element in M_2 will *reject*):
 - (a) If $X_{-1} \in M_1$, then $X_0 \in M_1$: *report M_1 as the cluster*
 - (b) If the cluster of X_{-1} is unknown, do a full IMH-step and *report the cluster of X_0*

If we choose k small enough so that $q(M_1) \approx \pi(M_1) \approx 1$, we can expect to end up in step 2a “most of the time”. In that case we only have to go back one time step to determine the likelihood cluster of the current draw. To determine that $X_{-1} \in M_1$ in step 3a, we simply repeat the procedure with a new candidate Q and a new random number $U \sim \text{Uniform}(0, 1)$ to see if we end up in step 2a. In the cases where a full (regular) IMH-step has to be done, we only report the *cluster* of X_0 in order not to bias the samples.

Second stage

Once the first stage is completed, we know to which cluster to restrict the simulations in the second stage. For the multistage coupler to be useful, it is crucial that we know an efficient way to draw from $\pi|M_1$. Usually, M_1 is a compact set. That information could help in finding a way to sample within that cluster. If the cluster from the first stage turns out to be M_2 , a possible way to simulate from $\pi|M_2$ is to reduce k and start another multistage coupler with M_2 as the state space.

Remark

In the context of model selection, for example, for the cases in Section 4.3 and Section 4.4 that were solved using an IMH-algorithm, a multistage coupler could be applied once an efficient way to sample from $\pi|M_1$ is found.

References

- Casella, G., Lavine, M., Robert, C., 2000. Explaining the perfect sampler. Working Paper 00-16, State University of New York at Stony Brook, Duke University, Durham.
- Corcoran, J., Schneider, U., 2001. Shift and scale coupling methods for perfect simulation. Submitted for publication. Preprint at: <http://amath.colorado.edu/faculty/corcoran/Papers/papers.html> .
- Corcoran, J., Tweedie, R., 2002. Perfect sampling from Independent Metropolis-Hastings Chains. *Journal of Statistical Planning and Inference* 104, 297–314.
- Fill, J., 1998. An interruptible algorithm for perfect sampling via Markov chains. *ANNAP* 8, 131–162.
- Foss, S., Tweedie, R., 1998. Perfect simulation and backward coupling. *Stochastic Models* 14, 187–203.
- Foss, S., Tweedie, R., Corcoran, J., 1998. Simulating the invariant measures of Markov chains using horizontal backward coupling at regeneration times. *Prob. Eng. Inf. Sci.* 12, 303–320.
- Häggström, O., van Liesholt, M., Møller, J., 1999. Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. *Bernoulli* 5, 641–659.
- Huang, Y., Djurić, P., 2002. Variable selection by perfect sampling. *Journal on Applied Signal Processing* 1, 38–45.
- Kendall, W., 1998. Perfect simulation for the area-interaction point process. In: Accardi, L., Heyde, C. (Eds.), *Probability Towards the Year 2000*. Springer, New York, pp. 218–234.
- Meng, X.-L., 2000. Towards a more general propp-wilson algorithm: Multistage backward coupling. *Monte Carlo Methods - Fields Institute Communications* 26, 85–93.
- Møller, J., 1999. Perfect simulation of conditionally specified models. *JRSSB* 61(1), 251–264.
- Murdoch, D., Green, P., 1998. Exact sampling from a continuous state space. *Scandinavian Journal of Statistics* 25, 483–502.
- Murdoch, D. J., 2000. Exact sampling for Bayesian inference: Unbounded state spaces. *Monte Carlo Methods - Fields Institute Communications* 26, 111–121.

- Propp, J., Wilson, D., 1996. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms* 9, 223–252.
- Raferty, A., Madigan, D., Hoeting, J., 1997. Bayesian model averaging for linear regression models. *JASA* 92, 179–191.
- Wilson, D., 2000. Layered multishift coupling for use in perfect sampling algorithms (with a primer on cftp). *Fields Institute Communications* 26, 141–176.