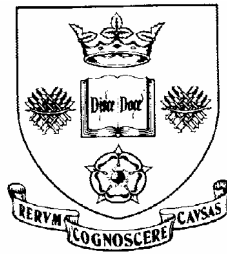# Sheffield Economic Research Paper Series

SERP Number: 2004001

Roberto León González

Data Augmentation in the Bayesian Multivariate Probit Model

January 2004

Department of Economics
University of Sheffield
9 Mappin Street
Sheffield
S1 4DT
United Kingdom
www.shef.ac.uk/economics

Data Augmentation in the Bayesian Multivariate Probit Model*

Roberto León González

Department of Economics

University of Sheffield

January 2004

Abstract

This paper is concerned with the Bayesian estimation of a Multivariate Probit model. In particular, this paper provides an algorithm that obtains draws with low correlation much faster than a pure Gibbs sampling algorithm. The algorithm consists in sampling some characteristics of slope and variance parameters marginally on the latent data. Estimations with simulated datasets illustrate that the proposed algorithm can be much faster than a pure Gibbs sampling algorithm. For some datasets, the algorithm is also much faster than the efficient algorithm proposed by Liu and Wu (1999) in the context of the univariate Probit model.

---

*This paper was circulated before as part of the discussion paper 09/02 of the Department of Economics at the University of York

1   Introduction.

Data augmentation (Tanner and Wong, 1988) was an important development in the field of Markov Chain Monte Carlo algorithms. When it is combined with the pioneer works of Metropolis et al. (1953) and Hastings (1970) it makes the Bayesian analysis of more complex models possible. Data augmentation consists in regarding latent and missing data as parameters to estimate. Although this introduces many more parameters, the conditional densities became much easier to sample from.

Although data augmentation facilitates the design of an algorithm, convergence might be slow due to high correlation between model parameters and latent data. Large autocorrelations imply that the chain moves slowly along the parameter space. Slow movement causes three types of problem. Firstly, a large number of iterations is needed for the chain to converge. In addition, a large number of iterations will be needed to obtain a representative sample of the posterior density. Furthermore, an even larger number of iterations are needed to be able to determine whether the chain has converged. A representative sample from the posterior density can be obtained after the chain has traveled along the parameter space just once. However, evidence of convergence requires that the chain has recovered the same region repeatedly. For these reasons it is advised (e.g. Raftery and Lewis 1992, Gilks and Roberts 1995) that when a chain is very slow an alternative algorithm must be designed.

The aim of this paper is to provide new tools to reduce the autocorrelations of the chain and hence enhance the reliability of the calculations in the Bayesian Multivariate Probit model. These strategies can potentially be applied to a wider range of Bayesian models.

The algorithm proposed in this paper combines both the Gibbs sampling and the Metropolis algorithm. Although data augmentation is used, some characteristics of slope and variance parameters are updated marginally on the latent data. These characteristics are sampled using a re-parameterisation of the posterior density. Estimation with a simulated dataset shows that the proposed algorithm can produce draws with low correlation much faster than a pure Gibbs sampling algorithm. In the particular case of a univariate Probit model, the proposed algorithm is also compared

with the PX-DA algorithm designed by Liu and Wu (1999). It is shown that the proposed algorithm substantially outperforms the PX-DA algorithm for some datasets.

The plan of the paper is as follows. Section 2 describes the Multivariate Probit model and the prior density. Section 3.1 and 3.2 outline the proposed algorithm for the Univariate and Multivariate Probit model, respectively. Section 4 uses simulated data to illustrate the performance of the proposed algorithm in comparison with a pure Gibbs sampling algorithm (Leon-Gonzalez, 2003) and the PX-DA algorithm (Liu and Wu, 1999). Section 5 draws some conclusions.

2 The Multivariate Probit Model

The Multivariate Probit model can be described as follows. Let Y_i be a vector of zeros and ones. Each component y_{it} of Y_i is determined by a continuous unobserved latent variable y_{it}^* generated according to the following process,

$$y_{it}^* = X_{it}\beta_t + e_{it} \qquad i = 1, ..., N \qquad t = 1, ..., T \qquad (1)$$

the vector $e_i = (e_{i1}, ..., e_{iT})^T$ is normally distributed with zero mean and covariance matrix $\Sigma = (\sigma_{jk})$. The binary variable y_{it} is equal to one if and only if $y_{it}^* \geq 0$, and is equal to zero otherwise. X_{it} is a $1 \times k_t$ vector of regressors and β_t is a vector of parameters.

The most common normalisation in the literature (e.g. Chib and Greenberg, 1998) is to fix:

$$\sigma_{11} = \sigma_{22} = ... = \sigma_{TT} = 1 \qquad (2)$$

However, with this normalisation Σ cannot be sampled directly from its conditional density, and a Metropolis step is necessary (Chib and Greenberg, 1998). It is possible to avoid this by following the normalisation restriction and the prior density proposed by Leon-Gonzalez (2003). In this way, it is possible to sample Σ directly from its conditional posterior, and hence it is possible to use a pure Gibbs sampling algorithm.

Thus, we use the following normalisation:

$$\sigma_{11} = \sigma_{22 \cdot 1} = ... = \sigma_{TT \cdot 12...(T-1)} = 1 \qquad (3)$$

where $\sigma_{TT \cdot 12 \ldots (T-1)} = Var\left(e_{iT}|e_{i(T-1)}, \ldots, e_{i1}\right)$.

Let the prior for Σ be an Inverted Wishart $IW_T\left(df_0, K_0\right)$ distribution conditional to restriction (3). That is, the kernel of the prior for a matrix Σ that verifies restriction (3) is:

$$|\Sigma|^{-df_0/2} \exp\left(-1/2 tr\left(\Sigma^{-1}K_0\right)\right) \tag{4}$$

It can be shown that normalisation (3) is equivalent to fixing the diagonal elements of the Cholesky decomposition of Σ equal to 1 (Leon-Gonzalez, 2003). Hence, normalisation (3) guarantees that the matrix Σ is positive definite. Hence, no further restriction holds on the parameters in Σ.

Let us assume that the vector of slope parameters $\left\{\beta_1^T, \beta_2^T, \ldots, \beta_T^T\right\}^T$ is a priori independent of Σ and follows a normal density with mean β_0 and variance-covariance matrix V_0.

The conditional posterior of Σ given parameters β and latent data $\{y_{it}^* : t = 1, \ldots, T\}_{i=1}^T$ is an inverted Wishart $IW_T\left(df, K\right)$ conditional to restriction (3). The parameters of this inverted Wishart are $df = df_0 + N$ and $K = K_0 + \sum_{i=1}^N e_i e_i^T$. A draw from this density can be obtained with a simple algorithm (Leon-Gonzalez 2003).

Note that the MCMC sample from the posterior density under normalisation (3) can be transformed to obtain a sample from the posterior of the parameters under normalisation (2) (Leon-Gonzalez, 2003). Hence, the use of normalisation (3) in the calculations does not preclude us from obtaining estimates according to restriction (2).

3 Data Augmentation in the Multivariate Probit Model.

For simplicity in the exposition, we first focus in the case of the probit model, that is $T = 1$. Section 3.2 considers the multivariate case.

3.1 The Univariate Probit.

The algorithm proposed in this section results from adding one step to the Gibbs sampling algorithm. The additional step proposes a move of model parameters that is carried out without conditioning on the latent data. In particular, slope parameters are multiplied by a random factor, and this random factor is sampled marginally on the latent data. The random factor is one variable from a re-parameterisation of the model.

Assuming that there are k_1 explanatory variables in the equation, let $\beta_1 = (\beta_{11}, ..., \beta_{1k_1})^T$, and consider the following re-parameterisation of the model,

$$\overline{\beta_1} = \left(\beta_{11}, \frac{\beta_{12}}{\beta_{11}}, \frac{\beta_{13}}{\beta_{11}}, ..., \frac{\beta_{1k_1}}{\beta_{11}}\right) = \left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$$
$$y_{i1}^* = X_{i11}\beta_{11} + X_{i12}\overline{\beta_{12}}\beta_{11} + X_{i13}\overline{\beta_{13}}\beta_{11} + ... + X_{i1k_1}\overline{\beta_{1k_1}}\beta_{11} + e_{i1}$$

The posterior distribution of $\overline{\beta_1}$, denoted by $\overline{\pi_M}(.)$, is derived from the posterior density of β_1 (π_M) in this way:

$$\overline{\pi_M}\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}|Y_1, ..., Y_N\right) = \pi_M\left(\beta_{11}, \overline{\beta_{12}}\beta_{11}, ..., \overline{\beta_{1k_1}}\beta_{11}|Y_1, ..., Y_N\right)|\beta_{11}|^{k_1-1}$$

where $\left|\beta_{11}^{k_1-1}\right|$ is the Jacobian of the transformation.

The proposed algorithm is:

Algorithm 1

Step 1: Sample $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ *conditional on* $(Y_1^*, ..., Y_N^*)$

Step 2: Sample β_{11} *conditional on* $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$

Step 3: Sample $(Y_1^*, ..., Y_N^*)$ *conditional on* $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$

Note that Algorithm 1 is the same as a Gibbs algorithm, with an additional step 2. Hence, it is just a Gibbs sampling algorithm in which all slope parameters (of the original parameterisation) might be multiplied by a random factor. Whenever a new value is drawn in the second step, all slope parameters move in the same direction. Step 2 accelerates the algorithm because it proposes a change of all parameters that is unconditional on the latent data. Conditioning on latent data makes the parameters move substantially more slowly.

The conditional distribution of $(Y_1^*, ..., Y_N^*)$ is the same as in the algorithm proposed by Chib and Greenberg (1998). The vector $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ can be sampled by

generating $(\beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ conditional on $(Y_1^*, ..., Y_N^*)$ and then transforming the variables in this way: $\left(\beta_{11}, \overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right) = (\beta_{11}, \beta_{12}/\beta_{11}, ..., \beta_{1k_1}/\beta_{11})$.

The conditional distribution of β_{11} given $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right)$ does not have a standard form. Hence, a metropolis step can be used to generate β_{11}. The proposal density for the Metropolis step could be a normal with mean and variance equal to the Maximum Likelihood estimation of β_{11}. Alternatively, β_{11} can be generated with a random walk. Step 2 can be repeated a number of times to increase the probability of accepting a new value. A more detailed description of the algorithm can be found in the Appendix.

3.2   The Multivariate Case

The algorithm for the multivariate case proposed in this section adds T steps to the Gibbs sampling algorithm. Each of these steps proposes to move slope and variance-covariance parameters for one equation jointly and without conditioning on the latent data for the corresponding equation. This move consists in multiplying the parameters by a random factor. This random factor is a variable from a re-parameterisation of the model and it is sampled marginally on the latent data.

Consider the following parameterisation of the multivariate probit model.

$$y_{it}^* = X_{it1}\beta_{t1} + X_{it2}\overline{\beta_{t2}}\beta_{t1} + X_{it3}\overline{\beta_{t3}}\beta_{t1} + ... + X_{itk_t}\overline{\beta_{tk_t}}\beta_{t1} + e_{it} \qquad t = 1, ..., T \quad (5)$$

where:

$$
\begin{aligned}
\overline{\beta_t} &= \left(\beta_{t1}, \frac{\beta_{t2}}{\beta_{t1}}, \frac{\beta_{t3}}{\beta_{t1}}, ..., \frac{\beta_{tk_t}}{\beta_{t1}}\right) = \left(\beta_{t1}, \overline{\beta_{t2}}, ..., \overline{\beta_{tk_t}}\right) & t = 1, ..., T \\
\overline{\Sigma_2} &= \left(\sigma_{12}/\left(\beta_{11}\beta_{21}\right), \sigma_{13}/\left(\beta_{11}\beta_{31}\right), ..., \sigma_{1T}/\left(\beta_{11}\beta_{T1}\right)\right) = \left(\overline{\sigma_{12}}, \overline{\sigma_{13}}, ..., \overline{\sigma_{1T}}\right) \\
\overline{\Sigma_3} &= \sigma_{23}/\left(\beta_{21}\beta_{31}\right), \sigma_{24}/\left(\beta_{21}\beta_{41}\right), ..., \sigma_{2T}/\left(\beta_{21}\beta_{T1}\right) = \left(\overline{\sigma_{23}}, \overline{\sigma_{24}}, ..., \overline{\sigma_{2T}}\right) \\
&\quad ... \\
\overline{\Sigma_T} &= \sigma_{(T-1)T}/\left(\beta_{T1}\beta_{(T-1)1}\right) = \overline{\sigma_{(T-1)T}}
\end{aligned}
$$

With this parameterisation, the covariance matrix of (e_i) is equal to:

$$
\Sigma = \begin{pmatrix}
\sigma_{11} & \overline{\sigma_{12}}\beta_{11}\beta_{21} & ... & \overline{\sigma_{1T}}\beta_{11}\beta_{T1} \\
\overline{\sigma_{12}}\beta_{11}\beta_{21} & \sigma_{22} & ... & \overline{\sigma_{2T}}\beta_{21}\beta_{T1} \\
& & ... & \\
\overline{\sigma_{1T}}\beta_{11}\beta_{T1} & \overline{\sigma_{2T}}\beta_{21}\beta_{T1} & ... & \sigma_{TT}
\end{pmatrix}
\qquad (6)
$$

where $(\sigma_{11}, \sigma_{22}, ..., \sigma_{TT})$ are determined by normalization (3) (e.g. $\sigma_{22} = 1 + (\overline{\sigma_{12}}\beta_{11}\beta_{21})^2$).

Let $\{(\beta_t)^n : t = 1, ..., T\}$, $(\Sigma)^n$ be the nth value of $\{\beta_t : t = 1, ..., T\}$, Σ in the proposed MCMC chain. The following algorithm describes how to obtain this value.

Algorithm 2.

Step 1: Sample $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}\right)$ conditional on $(Y_1^*, ..., Y_N^*, \Sigma)$.

Step 2: Sample $\left\{\overline{\Sigma_t} : t = 2, ..., T\right\}$ conditional on $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}, Y_1^*, ..., Y_N^*\right)$

Step 3: For $t = 1, ..., T$ do:

- *Generate β_{t1} conditional on $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$, $\left\{\overline{\beta_{tk}} : k \neq 1\right\}$, $\left\{\overline{\beta_j} : j \neq t\right\}$, $\left\{\overline{\Sigma_t} : t = 2, ..., T\right\}$.*

- *Sample $\{y_{it}^*\}^{i=1,...,N}$ conditional on the most recent values of $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}\right)$, $\left\{\overline{\Sigma_t} : t = 2, ..., T\right\}$ and $\{y_{ik}^* : k \neq t\}^{i=1,...,N}$*

Step 4: Fix

$$(\beta_t)^n = (\beta_{t1}, \beta_{t1}\overline{\beta_{t2}}, ..., \beta_{t1}\overline{\beta_{tk_t}}) \qquad for \quad t = 1, ..., T$$

$$(\Sigma)^n = \begin{pmatrix} \sigma_{11} & \overline{\sigma_{12}}\beta_{11}\beta_{21} & ... & \overline{\sigma_{1T}}\beta_{11}\beta_{T1} \\ \overline{\sigma_{12}}\beta_{11}\beta_{21} & \sigma_{22} & ... & \overline{\sigma_{2T}}\beta_{21}\beta_{T1} \\ & & ... & \\ \overline{\sigma_{1T}}\beta_{11}\beta_{T1} & \overline{\sigma_{2T}}\beta_{21}\beta_{T1} & ... & \sigma_{TT} \end{pmatrix}$$

where $(\sigma_{11}, \sigma_{22}, ..., \sigma_{TT})$ are determined by normalization (3)

The first part of Step 3 uses a Metropolis step to update the slope parameters and covariance terms (of the original parameterisation) jointly and marginally on the latent data. It can be repeated several times to increase the likelihood of acceptance of a new value. In the simulations of Section 4, the proposal density in the Metropolis step is a random walk. Alternatively, it is possible to specify a normal density centered on Maximum Likelihood estimates. These Maximum Likelihood estimates could be obtained before the MCMC chain is started from separate estimation of the T equations using univariate Probit models.

Step 1 is carried out by sampling first $(\beta_1, \beta_2, ..., \beta_T)$ as explained in Chib and Greenberg (1998) and transforming the values to obtain $\left(\overline{\beta_1}, \overline{\beta_2}, ..., \overline{\beta_T}\right)$.

Step 2 is carried out by sampling first Σ and transforming the values to obtain $\left\{\overline{\Sigma_t} : t = 2, ..., T\right\}$. The density of Σ is an inverted Wishart subject to restriction (3). This density can be sampled directly (Leon-Gonzalez 2003).

The latent data in the second part of Step 3 are sampled as described by Chib and Greenberg (1998): from truncated normals and according to equations (5) and (6).

4   Comparing the Performance of the Algorithms.

4.1   Univariate Probit Model.

This section compares Algorithm 1 with a standard Gibbs Sampling algorithm and the PX-DA algorithm (Liu and Wu, 1999). Algorithm 1 is implemented with step 2' repeated 4 times. 8400 observations for seven explanatory variables were generated independently from a standard normal distribution. Slope coefficients are:

$$\beta_{11} = 1, \beta_{12} = 2, \beta_{13} = 0.5, \beta_{14} = -0.2, \beta_{15} = -1, \beta_{16} = 0.8, \beta_{17} = 0.8$$

Table 4.1 shows the value of the highest correlation for lags 5, 10 and 20. Autocorrelations in the chain are calculated using 29000 iterations after discarding the first 1000 iterations. The initial value of parameters was equal to zero. Autocorrelations in Algorithm 1 are the lowest, being less than half the autocorrelations in the PX-DA algorithm. In the Gibbs sampling algorithm, 80 lags are necessary for the autocorrelations of all parameters to be below 0.1. The PX-DA algorithm achieves the same with 20 lags. Algorithm 1 needs the lowest amount of lags, 10, for all correlations to be below 0.1.

The highest correlation for all algorithms, except for Algorithm 1, correspond to β_{12}. As noted by Liu and Wu (1999), autocorrelations in the probit model increase with the absolute value of the coefficients. In contrast, the autocorrelation of β_{12} in Algorithm 1 is the lowest, and the highest correspond to parameter β_{14}, that has the smallest absolute value.

8

	Lag 5	Lag 10	Lag 20
Gibbs Algorithm	0.79	0.66	0.50
PX-DA algorithm	0.41	0.17	0.04
Algorithm 1	0.23	0.06	0.05

Table 1: Maximum Autocorrelation of the Parameters

Large auto-correlations make it more difficult to determine whether the chain has converged. With 29000 Gibbs sampling iterations, after discarding the first 1000 iterations, the Geweke test (1992) rejects the null hypothesis of convergence for 3 out of 7 parameters. With the same number of iterations, the test accepts the null hypothesis of convergence of all the parameters in the other two algorithms.

The Gibbs algorithm and the PX-DA algorithm have similar computation time. However, Algorithm 1 needs approximately double computing time with this implementation. Taking this into account, Algorithm 1 produces 2 draws with correlation smaller than 0.1 four times faster than a Gibbs Sampling algorithm. However, there is almost no advantage over the PX-DA algorithm, since similar values for correlations can be obtained with approximately the same computing time. However, as the following example shows, when slope parameters have a larger value the gains in autocorrelation clearly outweigh the losses in computation time.

A similar exercise is carried out, with the same number of observations, but letting the value of the parameters be:

$$\beta_{11} = 3, \beta_{12} = 3, \beta_{13} = 3, \beta_{14} = -3, \beta_{15} = -3, \beta_{16} = -3, \beta_{17} = 3$$

Table 2 shows the maximum autocorrelation of the parameters. The PX-DA algorithm needs at least 50 lags for all autocorrelations to be below 0.1. Algorithm 1 has all autocorrelations below 0.1 with just 5 lags. Hence, the substantial gains in smaller autocorrelations in Algorithm 1 more than compensate for the additional computation time per iteration. That is, with this dataset Algorithm 1 obtains two draws with low correlation five times faster than the PX-DA algorithm.

	Lag 5	Lag 10	Lag 30	Lag 40	Lag 50
PX-DA Algorithm	0.76	0.57	0.23	0.15	0.07
Algorithm 1	0.09	0.04	0.02	0.05	0.01

Table 2: Auto-correlations when parameters have a large value

4.2 Multivariate Probit Model.

This section compares the performance of Algorithm 2 with a standard Gibbs algorithm. Two versions of Algorithm 2 are implemented: in one the first part of step 3 is repeated 3 times (Algorithm 2a), and in the other it is carried out just once (Algorithm 2b).

The data was generated according to the following random-effects type process:

$$
\begin{aligned}
y_{it}^* &= 1 * x_{1i} + 2 * x_{2i} + 0.5 * x_{3i} - 0.2 * x_{4i} - 1 * x_{5i} + 0.8 * x_{6i} \\
&\quad + 0.8 * x_{7i} + u_i + e_{it} \\
i &= 1, ..., 1200 \quad t = 1, ..., 7
\end{aligned}
$$

where $(e_{i1}, ..., e_{iT})$ follows a $N(0, I)$, u_i follows a $N(0, 1)$, and it is independent of e_{it}. The regressors are invariant with t and are generated independently from a standard normal distribution. The prior for the slope parameters is a normal distribution with zero mean and covariance matrix equal to 10000I. The prior for the free parameters in Σ is a restricted inverted Wishart with $K_0 = I$ and $df_0 = 2 * T + 1 = 15$.

Hence, in this specification, 49 slope parameters plus 21 covariance parameters are estimated. For simplicity, only the autocorrelations of the 7 slope parameters in the first equation and 7 covariance parameters are analysed. Autocorrelations are calculated with 9000 iterations after discarding the first 1000.

Table 3 shows that autocorrelations for slope parameters vanishes more quickly in Algorithm 2. From Table 4, the Gibbs sampling algorithm has at least one covariance parameter with a correlation as high as 0.13 after 100 lags. In fact, the Gibbs algorithm needed 120 lags for the maximum autocorrelation to be below 0.1. In contrast, in Algorithms 2a and 2b the maximum correlation for covariance parameters vanishes before 30 and 40 lags.

The computing time per iteration in Algorithm 2a and 2b is 2.6 and 1.9 times larger than in the Gibbs algorithm, respectively. The gains in lower autocorrelations more than compensate for the extra computing time, since the number of iterations needed for the Gibbs autocorrelations to be below 0.1 is about 4 and 3 times the number of iterations needed in Algorithm 2a and 2b, respectively. In addition, unlike the Gibbs Sampling algorithm, the chains produced by Algorithm 2a and 2b passed the convergence test proposed by Heidelberg et al. (1983), hence further increasing the reliability of the calculations.

	Lag 10	Lag 20	Lag 30	Lag 40	Lag 50	Lag 100
Gibbs Algorithm	0.54	0.40	0.34	0.29	0.23	0.13
Algorithm 2a	0.29	0.13	0.08	0.04	0.04	0.007
Algorithm 2b	0.44	0.23	0.12	0.08	0.06	0.004

Table 3: Autocorrelations for covariance parameters

5   Conclusions

The motivation underlying the algorithms proposed in this paper is that a pure Gibbs sampling algorithm moves slowly due to sampling separately variables that are highly correlated. For this reason, the algorithms in this paper propose to sample characteristics of model parameters not conditioning on the latent data. This is

	Lag 10	Lag 20	Lag 30	Lag 40	Lag 50
Gibbs Algorithm	0.57	0.37	0.28	0.20	0.13
Algorithm 2a	0.33	0.17	0.09	0.06	0.04
Algorithm 2b	0.43	0.24	0.15	0.09	0.04

Table 4: Autocorrelations for slope parameters.

achieved by sampling from a re-parameterisation of the model. In this way, parameters in the original parameterisation are updated marginally on the latent data.

Using simulated data, the previous section shows that Algorithm 2 can produce two draws with low correlation faster than a pure Gibbs sampling algorithm. In the simpler case of the univariate Probit model, the proposed Algorithm 1 was compared with the efficient algorithm proposed by Liu and Wu (1999). It was shown that for some datasets Algorithm 1 is substantially much faster.

The type of re-parameterisations considered in this paper facilitates the updating of large numbers of parameters jointly and marginally on the latent data. This is potentially applicable to many models with complex likelihoods, where conventional MCMC algorithms fail to yield reliable calculations in reasonable time.

Appendix

Let $\left(\beta_{11}^n, \beta_{12}^n, ..., \beta_{1k_1}^n\right)$ denote the nth value of $(\beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ in the chain. If a random walk is used, the algorithm to generate the $(n+1)^{th}$ value of $(\beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ is:

Algorithm 1

Step 1. Sample $(\beta_{11}, \beta_{12}, ..., \beta_{1k_1})$ from a $N(\mu_p, V_p)$, where $\mu_p = V_p\left(\sum_{i=1}^N X_{i1}^T Y_i^* + V_0^{-1}\beta_0\right)$, and $V_p = \left(\sum_{i=1}^N X_{i1}^T X_{i1} + V_0^{-1}\right)^{-1}$.

Step 2. Fix $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right) = (\beta_{12}/\beta_{11}, ..., \beta_{1k_1}/\beta_{11})$. Generate a random scalar v from a distribution with density function $f(v)$. Fix

$$\beta_{11}^{n+1} = v\beta_{11}, \beta_{12}^{n+1} = v\beta_{11}\overline{\beta_{12}}, ..., \beta_{1k_1}^{n+1} = v\beta_{11}\overline{\beta_{1k_1}}$$

with probability

$$\gamma = \min\left\{\frac{L\left(Y|v\beta_{11}, v\beta_{12}, ..., v\beta_{1k_1}\right)\pi\left(v\beta_{11}, v\beta_{12}, ..., v\beta_{1k_1}\right)}{L\left(Y|\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right)\pi\left(\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right)}\frac{f(1/v)}{f(v)}\left|v^{k_1}\right|, 1\right\}$$

and fix

$$\beta_{11}^{n+1} = \beta_{11}, \beta_{12}^{n+1} = \beta_{11}\overline{\beta_{12}}, ..., \beta_{1k_1}^{n+1} = \beta_{11}\overline{\beta_{1k_1}}$$

with probability $(1-\gamma)$, where $L\left(Y|\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right)$ is the likelihood function:

$$L(Y|\beta_1) = \prod_{i=1}^N \left(\Phi\left(-X_{i1}\beta_1\right)^{1-Y_i}\left(1 - \Phi\left(-X_{i1}\beta_1\right)\right)^{Y_i}\right)$$

and $\pi\left(\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right)$ is the prior density.

Step 3. Sample Y_i^* from a truncated

$$N\left(X_{i11}\beta_{11}^{n+1} + X_{i12}\beta_{12}^{n+1} + ... + X_{i1k_1}\beta_{1k_1}^{n+1}, 1\right)$$

for all $i = 1, ..., N$.

Step 2 of the algorithm can be repeated a number of times to increase the likelihood of acceptance of a new value.

The function $f(v)$ might be centred at 1, so that new candidates are drawn from a distribution centred at the old value. In addition, it might be desirable to restrict $f(v)$

to positive values, hence forcing new candidates to have the same sign as the previous value. An inverted gamma would play this role.

Alternatively, the Metropolis step could use a normal density calibrated with the Maximum Likelihood estimates of β_{11}. If $\phi_N\left(x; \widehat{\beta_{11}}, \widehat{sd1}\right)$ is the density function of a $N\left(\widehat{\beta_{11}}, \widehat{sd1}\right)$ centered at the maximum likelihood estimates, then Step 2 can be implemented as:

- *Step 2'. Fix $\left(\overline{\beta_{12}}, ..., \overline{\beta_{1k_1}}\right) = (\beta_{12}/\beta_{11}, ..., \beta_{1k_1}/\beta_{11})$. Generate a random scalar v from a distribution with density function $\phi_N\left(v; \widehat{\beta_{11}}, \widehat{sd1}\right)$. Fix*

$$\beta_{11}^{n+1} = v, \beta_{12}^{n+1} = v\overline{\beta_{12}}, ..., \beta_{1k_1}^{n+1} = v\overline{\beta_{1k_1}}$$

  with probability

$$
\gamma' = \min\left\{1, \frac{L\left(Y|v, v\overline{\beta_{12}}, ..., v\overline{\beta_{1k_1}}\right) \pi\left(v, v\overline{\beta_{12}}, ..., v\overline{\beta_{1k_1}}\right)}{L\left(Y|\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right) \pi\left(\beta_{11}, \beta_{12}, ..., \beta_{1k_1}\right)} \frac{\phi_N\left(\beta_{11}^n; \widehat{\beta_{11}}, \widehat{sd1}\right)}{\phi_N\left(v; \widehat{\beta_{11}}, \widehat{sd1}\right)} \times \left|\left(\frac{v}{\beta_{11}}\right)^{k_1-1}\right|\right\}
$$

  and fix

$$\beta_{11}^{n+1} = \beta_{11}, \beta_{12}^{n+1} = \beta_{11}\overline{\beta_{12}}, ..., \beta_{1k_1}^{n+1} = \beta_{11}\overline{\beta_{1k_1}}$$

  with probability $(1 - \gamma')$.

References

Chib., S. and Greenberg, E. (1998) "Analysis of Multivariate Probit Models," *Biometrika* 85, 347-361.

Geweke, J., (1992) "Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments," in *Bayesian Statistics, 4* (ed. J. M. Bernardo, J. Berger, A. P. Dawid, and A. F. M. Smith) Oxford University Press, Oxford, 169-193.

Gilks, W. R., and Roberts, G. O. (1995) "Strategies for improving MCMC," In *Markov Chain Monte Carlo in Practice*, (ed. W.R. Gilks, S. Richardson and D.j. Spiegelhalter) Oxford University Press, Oxford, 89-114.

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

Heidelberg, P., and Welch, P. D. (1983) "Simulation Run Length Control in the Presence of and Initial Transient," *Operations Research*, 31, 1109-1144.

Leon-Gonzalez, R. (2003) "Sampling the Variance-Covariance Matrix in the Bayesian Multivariate Probit Model," *Discussion Paper SERP2003008, Department of Economics, University of Sheffield*

Liu, J. S., and Wu, Y. N., (1999) "Parameter Expansion for Data Augmentation," *Journal of the American Statistical Association*, 94, 448, 1264-1274.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) "Equation of state calculations by fast computing machine," *Journal of Chemical Physics* 21, 1087-91.

Raftery, A. L. and Lewis, S. (1992) "How many iterations in the Gibbs sampler?," In *Bayesian Statistics 4*, (ed. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith), 763-74. Oxford University Press, Oxford.

Tanner, T. A., and Wong, W. H. (1988) "The Calculation of Posterior Distributions by Data Augmentation," *Journal of the American Statistical Association* 82, 528-550.