A Corrected and More Efficient Suite of MCMC Samplers for the Multinomal Probit Model

Xiyun Jiao and David A. van Dyk *

Statistics Section, Department of Mathematics, Imperial College London

Abstract

The multinomial probit (MNP) model is a useful tool for describing discrete-choice data and there are a variety of methods for fitting the model. Among them, the algorithms provided by Imai and van Dyk (2005a), based on Marginal Data Augmentation, are widely used, because they are efficient in terms of convergence and allow the possibly improper prior distribution to be specified directly on identifiable parameters. Burgette and Nordheim (2012) modify a model and algorithm of Imai and van Dyk (2005a) to avoid an arbitrary choice that is often made to establish identifiability. There is an error in the algorithms of Imai and van Dyk (2005a), however, which affects both their algorithms and that of Burgette and Nordheim (2012). This error can alter the stationary distribution and the resulting fitted parameters as well as the efficiency of these algorithms. We propose a correction and use both a simulation study and a real-data analysis to illustrate the difference between the original and corrected algorithms, both in terms of their estimated posterior distributions and their convergence properties. In some cases, the effect on the stationary distribution can be substantial.

Keywords: Bayesian Analysis; Data Augmentations; Prior Distributions; Probit Models; Convergence

*Xiyun Jiao is a postgraduate student in Statistics, Department of Mathematics, Imperial College London, SW7 2AZ (Email: **x.jiao12@imperial.ac.uk**); Professor David A. van Dyk holds a Chair in Statistics, Department of Mathematics, Imperial College London (Email: **d.van-dyk@imperial.ac.uk**).

1 Introduction

The multinomial probit (MNP) model is widely used for describing discrete-choice data in social sciences and transportation studies. It is often preferred over the multinomial logit model because it does not assume independence of irrelevant alternatives; see, e.g., Hausman and Wise (1978) for details. Moreover, the MNP model has a strong connection with the multiperiod probit model, for which binary choices are observed over multiple time periods with correlated errors (McCulloch and Rossi, 1994).

The use of the MNP model was once restricted, because methods, like maximum likelihood estimates or simulated moments (McFadden, 1989), require evaluating high-dimensional normal integrals, which are typically intractable. More recently, advances in Bayesian simulations have boosted the development of Markov chain Monte Carlo (MCMC) algorithms for fitting the MNP model (e.g., McCulloch and Rossi (1994), Nobile (1998), McCulloch *et al.* (2000), Imai and van Dyk (2005a), and Burgette and Nordheim (2012)). These algorithms avoided evaluating multidimensional integrals, provided reliable model fitting, and thus revitalized the use of the MNP model in practice.

Current MCMC algorithms specify a set of latent Gaussian variables as augmented data, whose relative magnitudes determine the choices. Since the augmented model is not identifiable given the observations, a proper prior distribution is required to ensure that the posterior distribution is proper. McCulloch and Rossi (1994) advocate a Gibbs sampler which was the first feasible Bayesian approach to fitting the MNP model. In their specification, however, the prior distribution for the identifiable parameters is only determined as a byproduct (Imai and van Dyk, 2005a, henceforth IvD). An improvement of McCulloch and Rossi (1994) is the "hybrid Markov chain" introduced by Nobile (1998), which adds a Metropolis step to sample the unidentifiable parameters. McCulloch *et al.* (2000) propose another modification of McCulloch and Rossi (1994) which specifies a prior distribution directly on the identifiable parameters. IvD review these MCMC algorithms, compare their computational performance, and find that, first, Nobile (1998) can be less sensitive to starting values than McCulloch and Rossi (1994); second, although Nobile's method significantly improves the convergence of the overall chain, the gain seems to be primary for the unidentifiable parameter with only slight gain for the identifiable ones; and third, although McCulloch *et al.* (2000) solve the problem of prior specification, their algorithm can be less efficient in terms of convergence than either McCulloch and Rossi (1994) or Nobile (1998) (This final point was also noted by McCulloch *et al.* (2000) and Nobile (2000).) Moreover, IvD point out an error in Nobile's derivation which can alter its stationary distribution. Ironically, as we shall see, the algorithms of IvD also contain an error.

IvD develop new samplers based on the Marginal Data Augmentation (MDA) algorithm (Meng and van Dyk, 1999). The new algorithms are easy to implement because they only include draws from standard distributions. IvD demonstrate that first, their methods are at least as quick as the fastest methods in terms of convergence, and second, the model is specified in terms of possibly improper prior distributions that are set directly on the identifiable parameters, making the priors relatively easy to interpret. Because of their apparent advantages, IvD's algorithms have been widely used in practice to fit MNP models; see, e.g., Berrett and Calder (2012), Burgette and Nordheim (2012), Chaudoin (2014), Horiuchi *et al.* (2007), Hruschka (2007), Lu *et al.* (2012), Queralt (2012), Sinclair and Whitford (2013), Vincent *et al.* (2013), Zhang *et al.* (2008), etc. This success has been aided by a popular R package (*MNP*, Imai and van Dyk (2005b)).

Unfortunately, there are two errors in IvD's algorithms; both occur when sampling the variance-covariance matrix. First, IvD reparameterize the variables to facilitate the sampling of the variance-covariance matrix, and they make a mistake when transforming to the original parameterization. Second, when updating the variance-covariance matrix, a constraint on the matrix is overlooked. These errors can alter the stationary distribution and hence the fitted values and standard errors of the model parameters. They also can affect the efficiency of convergence.

Burgette and Nordheim (2012, henceforth BN) modify the model of IvD by changing the manner in which unidentifiability in the scale is addressed. In particular, they fix the trace of the variance-covariance matrix while IvD, like previous authors, fix the first diagonal element. BN's algorithm for sampling from the posterior distribution builds upon Algorithm 1 of IvD. Thus the two errors made by IvD also affect BN's algorithm. BN even make another mistake when updating the regression coefficient parameter, β. In this paper, we explain how to correct the errors in algorithms of both IvD and BN, and use both a simulation study and a real-data analysis to illustrate the difference between the original and the corrected algorithms in terms of their estimated posterior distributions and convergence properties. The corrections we propose will be implemented in the *MNP* R package.

The remainder of this paper is organized into five sections. We introduce the MNP model in Section 2. In Section 3, we present the original algorithms in IvD and BN, point out their errors, and provide the corrected algorithms. We also include a short review of MDA, which is used by all the algorithms we consider. In Sections 4 and 5, we use a simulation study and a real-data example, respectively, to illustrate the difference between the original and corrected algorithms. Conclusion and final remarks appear in Section 6.

2 Multinomial Probit Model

We consider a $(p+1)$-class multinomial model. Each observation is a binary $(p+1)$-vector, $d_i = (d_{i1}, \ldots, d_{i,(p+1)})$. We model d_i by conditioning on a latent multivariate normal variable, $U_i = \big(U_{i1}, \ldots, U_{i,(p+1)}\big)$; d_{ij} is one if U_{ij} is larger than all the other components of U_i. Specifically,

$$U_i \sim \mathrm{N}_{p+1}\left(X_i^0\beta, \Sigma^0\right) \text{ and } d_{ij} = \begin{cases} 1 & \text{if } U_{ij} = \max\{U_{i1}, \ldots, U_{i,(p+1)}\} \\ 0 & \text{otherwise} \end{cases}, \quad \text{for } i = 1, \ldots, n, \tag{1}$$

where X_i^0 is a $((p+1) \times q)$ matrix of known covariates, β is a q-vector of unknown parameters, and Σ^0 is a $((p+1) \times (p+1))$ unknown variance-covariance matrix.

Model (1) is unidentifiable because shifting U_i by any constant or rescaling U_i by any positive constant, does not alter the distribution of d_i. To avoid this, IvD and BN both follow McCulloch and Rossi (1994), by expressing each U_{ij} relative to a base category (e.g., $U_{i,(p+1)}$), and obtain the new latent variable, $W_i = (W_{i1}, \ldots, W_{ip})$, where $W_{ij} = U_{ij} - U_{i,(p+1)}$. The distribution of W_i is still multivariate normal, that is,

$$W_i \sim \mathrm{N}_p(X_i\beta, \Sigma), \tag{2}$$

where $X_i = PX_i^0$ and $\Sigma = P\Sigma^0 P^{\mathrm{T}}$ with $P = [I_p, -J]$, with I_p a $(p \times p)$ identity matrix and J a column p-vector of ones. For simplicity, we collapse d_i into Y_i, which is an integer in $\{0, \ldots, p\}$, defined as

$$Y_i = \begin{cases} 0 & \text{if } \max\{W_{i1}, \ldots, W_{ip}\} < 0 \\ k & \text{if } W_{ik} = \max\{0, W_{i1}, \ldots, W_{ip}\} \end{cases}, \quad \text{for } i = 1, \ldots, n. \tag{3}$$

To ensure identifiability, we must also set the scale. IvD adopt the standard solution of McCulloch and Rossi (1994); they set the first diagonal element of Σ to one, i.e., $\sigma_{11}^2 = 1$. BN propose a different solution; they fix the trace of the variance-covariance matrix, i.e., $\mathrm{trace}(\Sigma) = p$. They argue that the trace restriction is a better choice from three reasons. First, the trace restriction makes it possible to specify a symmetric prior for Σ that is invariant to permutations of rows and columns. Second, when using the variance-element restriction (as in IvD), the estimated predicted choice probabilities under the posterior distribution can vary largely with the choice of the category corresponding to the unit variance. The trace-restricted fits tend to be intermediate among the results of the p possible variance-element restricted fits. Third, the trace restriction yields marginal posterior distributions that are easier to interpret.

To overcome difficulties stemming from the constraint, $\sigma_{11}^2 = 1$, on the variance-covariance matrix,

motivated by McCulloch and Rossi (1994), IvD set $\tilde{W}_i = \alpha W_i$, for $i = 1, \ldots, n$, where $\alpha > 0$. Then $\tilde{W}_i \sim \mathrm{N}_p(X_i\tilde{\beta}, \tilde{\Sigma})$, where $\tilde{\beta} = \alpha\beta$ and $\tilde{\Sigma} = \alpha^2\Sigma$. Because $\tilde{\Sigma}$ can be any positive-definite matrix, IvD specify an inverse-Wishart prior distribution, $\tilde{\Sigma} \sim \text{Inv-Wishart}(\nu, \tilde{S})$. After transforming to $\alpha^2 = \tilde{\sigma}_{11}^2$ and $\Sigma = \tilde{\Sigma}/\tilde{\sigma}_{11}^2$, the implied prior distribution on (α^2, Σ) is

$$\alpha^2|\Sigma \sim \alpha_0^2 \text{trace}(S\Sigma^{-1})/\chi_{\nu p}^2, \text{ and } p(\Sigma) \propto |\Sigma|^{-(\nu+p+1)/2}[\text{trace}(S\Sigma^{-1})]^{-\nu p/2} I\{\sigma_{11}^2 = 1\}, \tag{4}$$

where $S = \tilde{S}/\alpha_0^2$ and the first diagonal element of S is one; I is an indicator function which equals one when the condition in the brackets is satisfied, and zero otherwise. They also specify a normal prior distribution for β, $\beta \sim \mathrm{N}_q(\beta_0, A)$. For simplicity, we set $\beta_0 = 0$. BN adopt the same strategy for setting their prior distribution in the context of the constraint, $\text{trace}(\Sigma) = p$. In particular, their implied prior distribution for (α^2, Σ) is almost the same as IvD except that

$$p(\Sigma) \propto |\Sigma|^{-(\nu+p+1)/2}[\text{trace}(S\Sigma^{-1})]^{-\nu p/2} I\{\text{trace}(\Sigma) = p\}, \tag{5}$$

where $\text{trace}(S) = p$. They use the same prior distribution as IvD for β, i.e., $\beta \sim \mathrm{N}_q(0, A)$. As IvD state, this choice of prior distribution allows both informative and diffuse priors for unknown parameters while maintaining simplicity and efficiency of the algorithms.

3 MDA Algorithms for Fitting MNP Models

3.1 Marginal Data Augmentation

The algorithms of IvD and BN are all based on the method of MDA. To describe and correct the errors in these algorithms, we briefly review MDA. First, denoting (β, Σ) by θ, the Data Augmentation (DA) algorithm (Tanner and Wong, 1987) is designed to sample from the posterior distribution, $p(\theta, W|Y)$, by updating from $p(W|\theta, Y)$ and $p(\theta|W, Y)$ iteratively. In this section, we regard Y, θ and W as generic observed data, unknown parameter of interest, and latent variables, respectively.

Although easy to implement, the DA algorithm can be slow to converge. The MDA algorithm (Meng and van Dyk, 1999) improves the convergence rate of a standard DA algorithm by expanding its state space. Specifically, MDA introduces a working parameter, α, into the augmented-data model $p(W, Y|\theta)$; α is not identifiable under the observed-data model $p(Y|\theta)$. An MCMC sampler is run on the expanded

model $p(\tilde{W}, Y|\theta, \alpha)$, which is designed to maintain $p(Y|\theta)$ as its marginal distribution, that is,

$$\int p(\tilde{W}, Y|\theta, \alpha)\mathrm{d}\tilde{W} = p(Y|\theta). \tag{6}$$

A general method for introducing α into an augmented-data model is to use a one-to-one mapping,

$$\tilde{W} = \mathcal{F}_\alpha(W), \text{ for any given } \alpha, \tag{7}$$

which is differentiable when W is continuous. For each $\mathcal{F}$, there typically exists a value α_0 such that $\mathcal{F}_{\alpha_0}$ is an identity function, $\mathcal{F}_{\alpha_0}(W) = W$. With this construction, the MDA algorithm proceeds by iterating

$$\text{Step 1: } (\tilde{W}^{(t+1)}, \alpha^\star) \sim p(\tilde{W}, \alpha|\theta^{(t)}, Y), \tag{8}$$

$$\text{Step 2: } (\theta^{(t+1)}, \alpha^{(t+1)}) \sim p(\theta, \alpha|\tilde{W}^{(t+1)}, Y).$$

Note that in the sampler in (8), α is sampled in both steps and its first update is not part of the final output. We define such updates as *intermediate quantities* and indicate them with superscript "$\star$". The sampler in (8) is a *collapsed DA sampler* (Liu *et al.*, 1994), since its two steps can be considered as sampling $\tilde{W}$ and θ with α integrated out. In this regard, the sampler in (8) is equivalent to a standard DA sampler constructed for the conditional distributions of $p(\tilde{W}, \theta|Y)$. Thus the marginal Markov chain, $\{\theta^{(t)}, t = 0, 1, \dots\}$, produced by the sampler in (8) is reversible with $p(\theta|Y)$ as its stationary distribution. Collapsing α out increases the (expected) variance of the conditional distributions sampled in (8). This enables bigger jumps and faster convergence, see Meng and van Dyk (1999) and van Dyk and Meng (2001) for more details.

3.2 Errors in Algorithms and the Corrections

We refer to Algorithms 1 and 2 of IvD as Algorithms 1.1 and 2.1. This allows us to clearly number the corrected versions of these algorithms. Similarly, we refer to the algorithm of BN as Algorithm 3.1. Algorithm 1.1 is displayed here and Algorithms 2.1 and 3.1 in Appendices A and B.

To obtain posterior samples under the MNP model, Algorithm 1.1 proceeds by sampling iteratively from $p(\alpha^2, \tilde{W}|Y, \beta, \Sigma)$, $p(\alpha^2, \beta|Y, \tilde{W}, \Sigma)$ and $p(\alpha^2, \Sigma|Y, \tilde{W} - \alpha X\beta, \beta)$. The first of these draws is obtained via a sequence of conditional draws, see Step 1(b) of Algorithm 1.1. Note that this algorithm marginalizes α out in each step. Algorithm 2.1 proceeds by sampling iteratively from $p(\alpha^2, \tilde{W}|Y, \beta, \Sigma)$, $p(\alpha^2, \Sigma|Y, \tilde{W} - \alpha X\beta, \beta)$ and $p(\beta|Y, W, \Sigma)$, again using a sequence of conditional draws for updating $\tilde{W}$. Algorithm 2.1

Algorithm 1.1

Step 0: Initialize parameters $t = 0$, $\beta^{(0)}$, $\alpha^{(0)}$, $\Sigma^{(0)}$ and $W^{(0)}$.

while $t < T$ **do**

Step 1: Update $\left((\alpha^2)^\star, \tilde{W}^\star\right)$ via $p(\alpha^2, \tilde{W}|Y, \beta^{(t)}, \Sigma^{(t)})$ by

(a) sampling $(\alpha^2)^\star$ from $p(\alpha^2|\Sigma^{(t)})$: $(\alpha^2)^\star \sim \alpha_0^2 \text{trace}\left(S{\Sigma^{(t)}}^{-1}\right)/\chi^2_{\nu p}$;

(b) sampling $\tilde{W}^\star$ from $p(\tilde{W}^\star|Y, (\alpha^2)^\star, \beta^{(t)}, \Sigma^{(t)})$:

for $i := 1, \ldots, n$ **do**

for $k := 1, \ldots, p$ **do**

sampling $W^\star_{ik}$ from $p(W_{ik}|Y_i, W^\star_{i,-k}, \beta^{(t)}, \Sigma^{(t)})$: $W^\star_{ik} \sim \text{TN}(\mu_{ik}, \tau^2_{ik})$, see Appendix C for details;

end for

Set $\tilde{W}^\star_i = \alpha^\star W^\star_i$.

end for

Step 2: Update $\left((\alpha^2)^\star, \beta^{(t+1)}\right)$ via $p(\alpha^2, \beta|Y, \tilde{W}^\star, \Sigma^{(t)})$ by

(a) sampling $(\alpha^2)^\star$ from $p(\alpha^2|Y, \tilde{W}^\star, \Sigma^{(t)})$:

$$(\alpha^2)^\star \sim \frac{\sum_{i=1}^n (\tilde{W}^\star_i - X_i\hat{\beta})^{\text{T}}{\Sigma^{(t)}}^{-1}(\tilde{W}^\star_i - X_i\hat{\beta}) + \hat{\beta}^{\text{T}}A^{-1}\hat{\beta} + \text{trace}\left(\tilde{S}{\Sigma^{(t)}}^{-1}\right)}{\chi^2_{(n+\nu)p}},$$

where $\hat{\beta} = \left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t)}}^{-1}X_i + A^{-1}\right)^{-1}\left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t)}}^{-1}\tilde{W}^\star_i\right)$;

(b) sampling $\tilde{\beta}^\star$ from $p(\tilde{\beta}|Y, \tilde{W}^\star, (\alpha^2)^\star, \Sigma^{(t)})$:

$$\tilde{\beta}^\star \sim \text{N}_q\left[\hat{\beta}, (\alpha^2)^\star\left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t)}}^{-1}X_i + A^{-1}\right)^{-1}\right],$$

and setting $\beta^{(t+1)} = \tilde{\beta}^\star/\alpha^\star$.

Step 3: Update $\left((\alpha^2)^{(t+1)}, \Sigma^{(t+1)}\right)$ via $p(\alpha^2, \Sigma|Y, \tilde{W}^\star, \beta^{(t+1)})$ by

(a) sampling $\tilde{\Sigma}^\star$ from $p(\tilde{\Sigma}|Y, Z, \beta^{(t+1)})$:

$$\tilde{\Sigma}^\star \sim \text{Inv-Wishart}\left(n + \nu, \sum_{i=1}^n Z_i Z_i^{\text{T}}\right),$$

where $Z_i = \tilde{W}^\star_i - \alpha^\star X_i\beta^{(t+1)}$;

(b) setting $\alpha^{(t+1)} = \tilde{\sigma}^\star_{11}$, $\Sigma^{(t+1)} = \tilde{\Sigma}^\star/\left(\alpha^{(t+1)}\right)^2$, and $W^{(t+1)} = \tilde{W}^\star/\alpha^{(t+1)}$.

return $\beta^{(t+1)}$, $\Sigma^{(t+1)}$ and $W^{(t+1)}$

$t + 1 \leftarrow t$

end while

does not marginalize α out when sampling β. Algorithm 3.1 is an adaption of Algorithm 1.1. The only difference occurs in Step 3 when sampling (α^2, Σ). In Algorithm 1.1, α^2 is set to the first element of $\tilde{\Sigma}$ in Step 3(b), while it is set to $\text{trace}(\tilde{\Sigma})/p$ in Step 3(b) of Algorithm 3.1.

Unfortunately, there are two errors in these algorithms, which may severely alter their stationary distributions, fitted values, and convergence properties. In Algorithm 1.1, both errors are in Step 3. The first is rather simple. The transformation from $(Z, \beta^{(t+1)}, \alpha^{(t+1)}, \tilde{\Sigma}^\star)$ to the original parameterization $(W^{(t+1)}, \beta^{(t+1)}, \alpha^{(t+1)}, \Sigma^{(t+1)})$ should involve setting

$$W_i^{(t+1)} = \left(Z_i + \alpha^{(t+1)}X_i\beta^{(t+1)}\right)/\alpha^{(t+1)}, \text{ for } i = 1, \ldots, n, \tag{9}$$

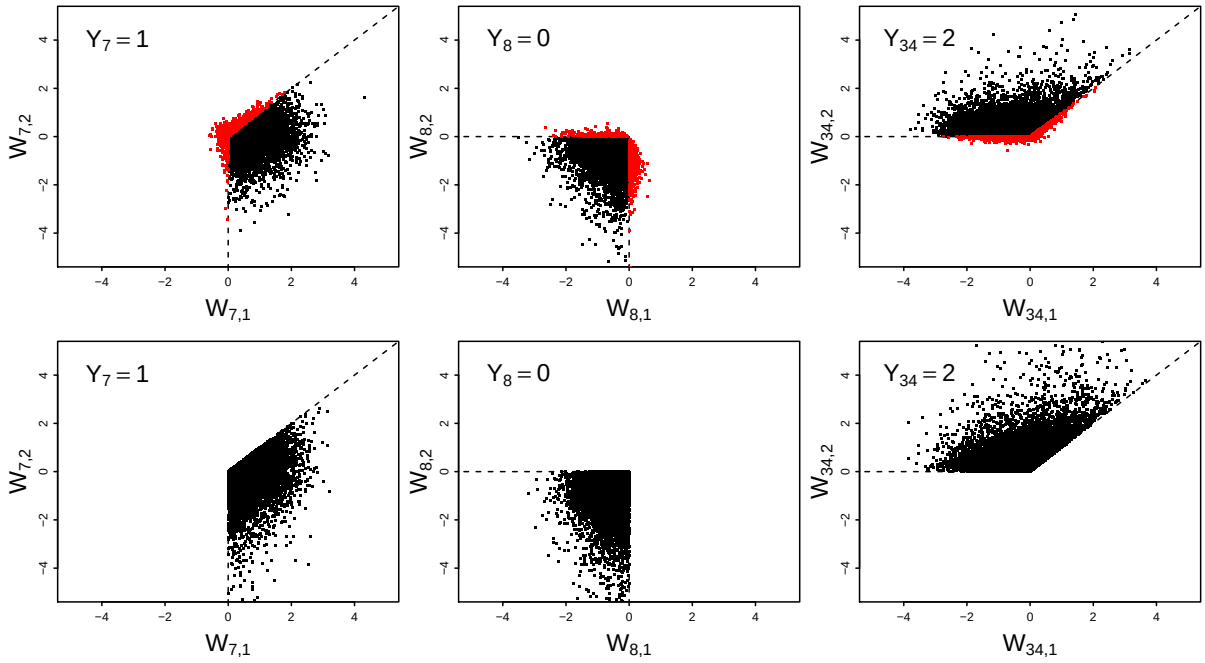

Figure 1: The posterior samples of W_7, W_8, and W_{34} obtained with Algorithms 1.2 and 1.3 appear in the first and second rows, respectively. The samples from Algorithm 1.2 not adhering to the constraint (10) are plotted in red.

instead of $W_i^{(t+1)} = \tilde{W}_i^\star/\alpha^{(t+1)}$, see Step 3(b). The correct inverse transformation is necessary to guarantee that the joint stationary distribution of $(W^{(t+1)}, \beta^{(t+1)}, \alpha^{(t+1)}, \Sigma^{(t+1)})$ is the target posterior distribution.

The second problem is more subtle. When sampling $\tilde{\Sigma}^\star$ while conditioning on Y, Z, and $\beta^{(t+1)}$, Algorithm 1.1 uses Inv-Wishart$(n+\nu, \sum_{i=1}^n Z_i Z_i^{\mathrm{T}})$, see Step 3(a). This however ignores a constraint on $\tilde{\Sigma}^\star$ imposed by Y and the current value of Z and β. This constraint is on the first diagonal element of $\tilde{\Sigma}^\star$, i.e., $(\tilde{\sigma}_{11}^\star)^2$. In particular, if we set $\tilde{Z}_i(\tilde{\sigma}_{11}^\star) = Z_i + (\tilde{\sigma}_{11}^\star) X_i \beta^{(t+1)}$, for $i = 1, \ldots, n$, the updated value of $\tilde{\sigma}_{11}^\star$ must satisfy

$$\begin{cases} \max\left\{\tilde{Z}_{i1}(\tilde{\sigma}_{11}^\star), \ldots, \tilde{Z}_{ip}(\tilde{\sigma}_{11}^\star)\right\} < 0 & \text{if } Y_i = 0 \\ \max\left\{0, \tilde{Z}_{i1}(\tilde{\sigma}_{11}^\star), \ldots, \tilde{Z}_{ip}(\tilde{\sigma}_{11}^\star)\right\} = \tilde{Z}_{ik}(\tilde{\sigma}_{11}^\star) & \text{if } Y_i = k \end{cases}, \quad \text{for} \quad i = 1, \ldots, n. \tag{10}$$

Thus, the conditional distribution of $\tilde{\Sigma}^\star$ given Y, Z, and $\beta^{(t+1)}$ in Step 3(a) should be a constrained inverse-Wishart distribution.

To illustrate the effect of the two corrections to Algorithm 1.1, we compare it with two new algorithms:

Algorithm 1.2: This is a partial correction to Algorithm 1.1. The only difference is that Algorithm 1.2 transforms $(Z, \beta^{(t+1)}, \alpha^{(t+1)}, \tilde{\Sigma}^\star)$ to $(W^{(t+1)}, \beta^{(t+1)}, \alpha^{(t+1)}, \Sigma^{(t+1)})$ using (9) in Step 3(b).

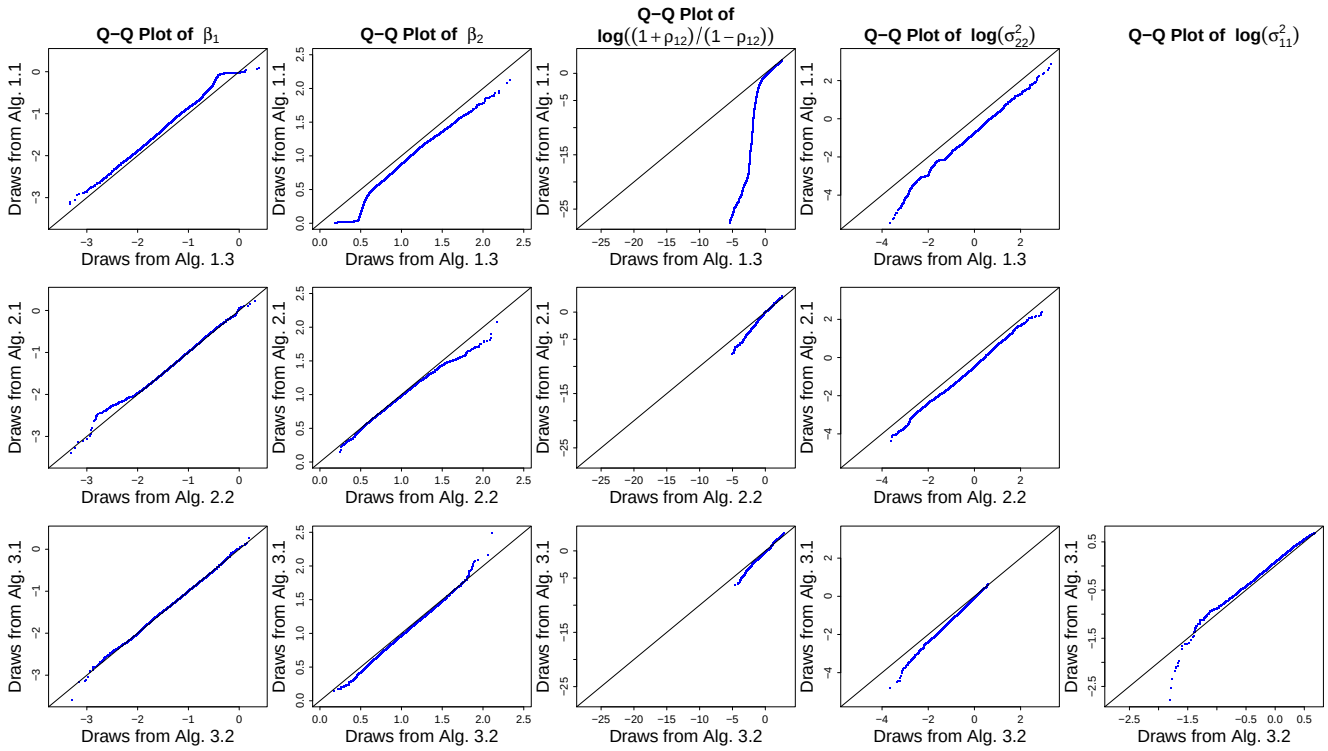

Figure 2: Quantile-quantile plots for comparing posterior draws from different algorithms in the simulation study. The columns correspond to five parameters, i.e., β_1, β_2, $\log\left(\frac{1+\rho_{12}}{1-\rho_{12}}\right)$, $\log(\sigma_{22}^2)$, and $\log(\sigma_{11}^2)$. The first row compares draws from Algorithms 1.1 and 1.3, the second row compares draws from Algorithms 2.1 and 2.2, and the last row compares draws from Algorithms 3.1 and 3.2.

Algorithm 1.3: This algorithm completely corrects Algorithm 1.1. In particular, Steps 0, 1, and 2 of Algorithm 1.3 are the same as Algorithm 1.1. In Step 3(a), however, Algorithm 1.3 updates $\tilde{\Sigma}^\star$ by sampling from a constrained inverse-Wishart distribution, that is,

$$\tilde{\Sigma}^\star \sim \text{Inv-Wishart}\left(n+\nu, \sum_{i=1}^{n} Z_i Z_i^{\mathrm{T}}\right) \text{ subject to the constraint in (10).}$$

This is accomplished by simple rejection sampling; we iteratively sample from the unconstrained inverse-Wishart distribution until (10) is satisfied. Finally, in Step 3(b), Algorithm 1.3 sets $\alpha^{(t+1)} = \tilde{\sigma}_{11}^\star$, $\Sigma^{(t+1)} = \tilde{\Sigma}^\star / \left(\alpha^{(t+1)}\right)^2$, and $W_i^{(t+1)} = (Z_i + \alpha^{(t+1)} X_i \beta^{(t+1)})/\alpha^{(t+1)}$.

Algorithms 2.1 and 3.1 are adaptions of Algorithm 1.1. Thus, both corrections affect these algorithms as well. The corrected versions of Algorithms 2.1 and 3.1 are called Algorithms 2.2 and 3.2 respectively. See Appendices A and B for details of these algorithms.

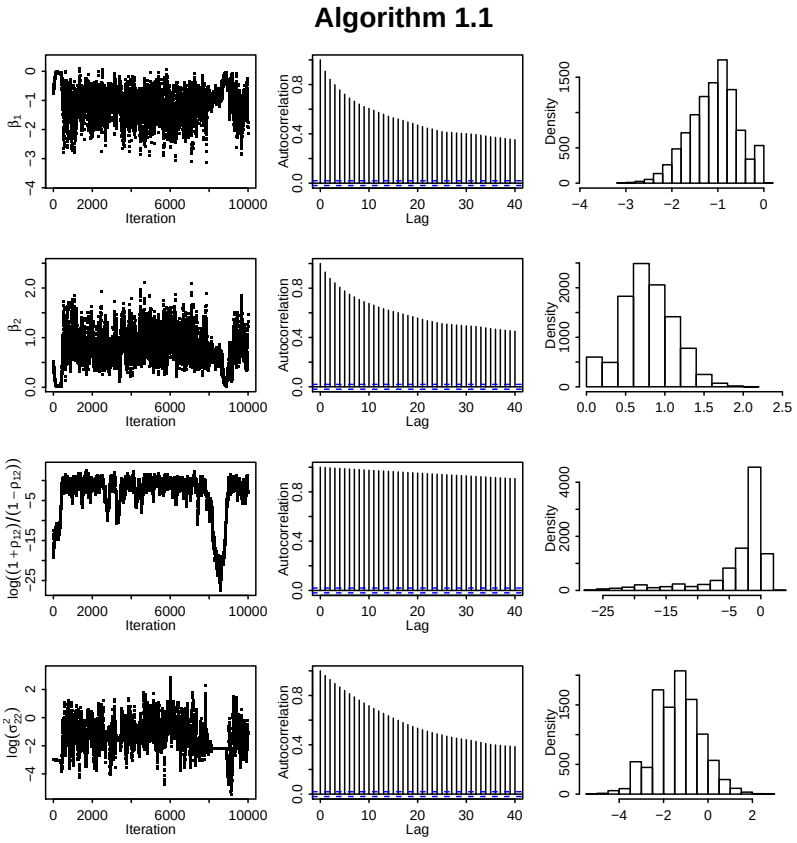

Figure 3: The sampling results of Algorithm 1.1 for the simulation study. The columns correspond to trace plots, autocorrelation plots, and histograms. The rows correspond to four parameters: β_1, β_2, $\log\left(\frac{1+\rho_{12}}{1-\rho_{12}}\right)$, and $\log(\sigma_{22}^2)$.

4 Simulation Study

We use a simulation study to illustrate the differences in the convergence properties of Algorithms 1.1, 1.2, and 1.3, Algorithms 2.1 and 2.2, and Algorithms 3.1 and 3.2. We set $n = 50$, $p = 2$, $q = 2$, $\beta = (-\sqrt{2}, 1)$, $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$. For $X_i = \begin{pmatrix} X_{i1,1} & X_{i1,2} \\ X_{i2,1} & X_{i2,2} \end{pmatrix}$, we sample $X_{ij,1}$ $(j = 1, 2)$ from a uniform distribution on $(-0.5, 0.5)$ for $i = 1, \ldots, 25$, on $(0.4, 1.5)$ for $i = 26, \ldots, 50$, and sample $X_{ij,2}$ $(j = 1, 2)$ from a uniform distribution on $(-1, 1)$ for $i = 1, \ldots, 25$, on $(0.8, 3)$ for $i = 26, \ldots, 50$. We specify the prior distribution of Σ and α^2 as in Section 2, with $\nu = p$, $\alpha_0^2 = \nu$, and $S = \text{Diag}(1, 1)$, and for β, as $\beta \sim \text{N}_q[0, \text{Diag}(100, 100)]$. For each algorithm, we run a chain of length 15,000 and discard the first 5,000 draws.

Figure 1 presents the posterior samples of W_7, W_8, and W_{34} obtained with Algorithms 1.2 and 1.3 respectively. The draws obtained with Algorithm 1.2 that do not adhere to the constraint (10) are colored in red, which illustrates the second problem of Algorithm 1.1 described in Section 3.2. Such draws are rejected in Step 3(a) of Algorithm 1.3.

Most importantly, Algorithms 1.1 (or 1.2), 2.1, and 3.1 do not return draws from the target poste-

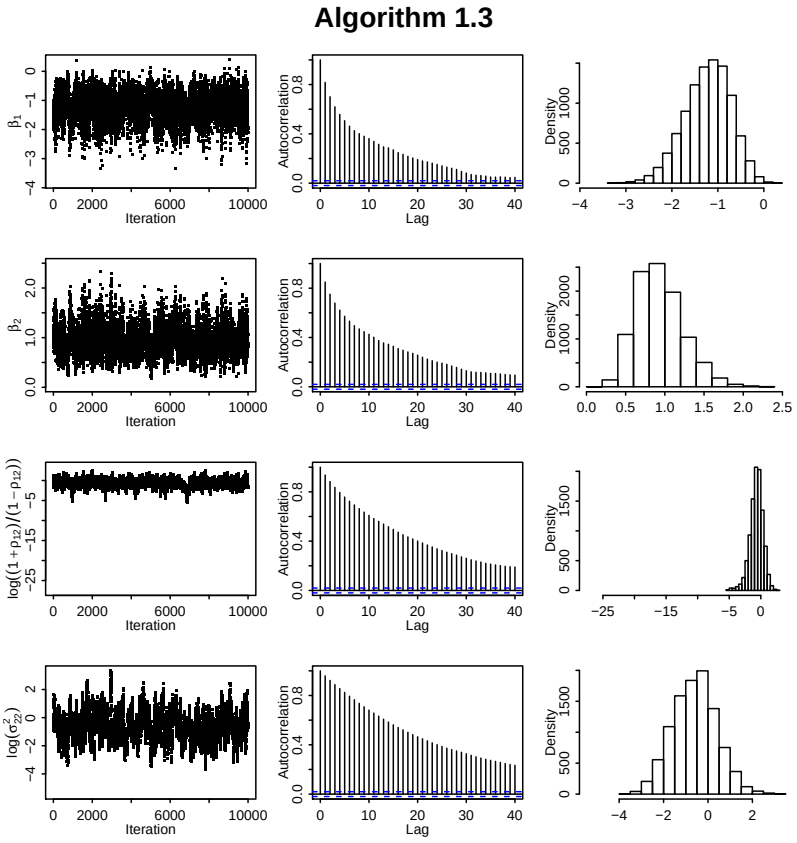

Figure 4: The sampling results of Algorithm 1.3 for the simulation study. The columns correspond to trace plots, autocorrelation plots, and histograms. The rows correspond to four parameters: β_1, β_2, $\log\left(\frac{1+\rho_{12}}{1-\rho_{12}}\right)$, and $\log(\sigma_{22}^2)$.

rior distribution. Figure 2 shows the quantile-quantile plots of parameters to compare the stationary distributions of original and corrected algorithms. The first row of Figure 2 compares Algorithms 1.1 and 1.3. The distributions of β differ slightly for the two algorithms, while the distributions of Σ differ significantly, especially the correlation parameter, $\rho_{12} = \sigma_{12}/(\sigma_{11}\sigma_{22})$. For Algorithms 1.2 and 1.3 (not shown), the distributions of β are again similar, while the distributions of Σ again differ, but not as severely as Algorithms 1.1 and 1.3. The second row shows the quantile-quantile plots that compare Algorithms 2.1 and 2.2. The distributions of both β and Σ are slightly different for the two algorithms. The last row of Figure 2 compares Algorithms 3.1 and 3.2. The distributions of β are rather similar for the two algorithms, while the distributions of Σ are different, particularly ρ_{12} and σ_{22}^2.

Figures 3 and 4 show the sampling results of Algorithms 1.1 and 1.3 respectively. The columns in both figures correspond to trace plots, autocorrelation plots, and histograms. The rows correspond to four parameters, namely, β_1, β_2, $\log\left(\frac{1+\rho_{12}}{1-\rho_{12}}\right)$, and $\log(\sigma_{22}^2)$. Algorithm 1.3 has better convergence properties than Algorithm 1.1 for all the four parameters in terms of mixing and autocorrelation. The convergence of

	Alg. 1.1	**Alg. 1.2**	**Alg. 1.3**	**Alg. 2.1**	**Alg. 2.2**	**Alg. 3.1**	**Alg. 3.2**
β_1	0.693	1.592	1.610	1.575	2.111	1.935	2.573
β_2	0.443	1.584	1.305	1.384	1.342	2.492	3.413
$\log\left(\sigma_{11}^2\right)$	-	-	-	-	-	1.445	1.351
$\log\left(\frac{1+\rho_{12}}{1-\rho_{12}}\right)$	0.060	0.625	0.865	0.782	1.070	0.768	1.163
$\log\left(\sigma_{22}^2\right)$	0.506	1.334	0.703	1.474	0.823	1.261	1.216

Table 1: Effective sample size per second for each of five parameters, i.e., β_1, β_2, $\log\left(\sigma_{11}^2\right)$, $\log\left(\frac{1+\rho_{12}}{1-\rho_{12}}\right)$, and $\log\left(\sigma_{22}^2\right)$ in the simulation study, as obtained with Algorithms 1.1–1.3, Algorithms 2.1–2.2, and Algorithms 3.1–3.2 respectively.

Algorithm 1.2 is better than Algorithm 1.1, but not as good as Algorithm 1.3. Moreover, Algorithms 2.2 and 3.2 have slightly better convergence properties than Algorithms 2.1 and 3.1 respectively. We omit the corresponding plots for Algorithms 1.2, 2.1, 2.2, 3.1, and 3.2 to save space.

To further compare the convergence properties of these algorithms, we compute the *effective sample size* (ESS), defined by

$$\mathrm{ESS}(\theta) = \frac{T}{1 + 2\sum_{t=1}^{\infty} \rho_t(\theta)}, \tag{11}$$

where T is the total posterior sample size and $\rho_t(\theta)$ is the lag-t autocorrelation of the parameter θ. ESS gives an estimate of the equivalent number of independent iterations that a Markov chain represents, and it indicates how well the chain mixes, see Kass *et al.* (1998) and Liu (2001). We use the function "effectiveSize" in the R package *coda* to calculate ESS. To account for the required CPU time, we compare ESS per second of these algorithms. The larger the value, the more efficient is the algorithm. The first three columns in Table 1 present the ESS per second for each parameter for Algorithms 1.1–1.3, respectively. The fourth and fifth columns correspond to Algorithms 2.1 and 2.2, and the last two columns correspond to Algorithms 3.1 and 3.2. We find that in terms of ESS per second, Algorithm 1.3 is more efficient than Algorithm 1.1 even though it is more computationally demanding per iteration, and it performs similarly to Algorithm 1.2. Algorithm 2.2 performs similarly to Algorithm 2.1, and Algorithm 3.2 performs slightly better than Algorithm 3.1.

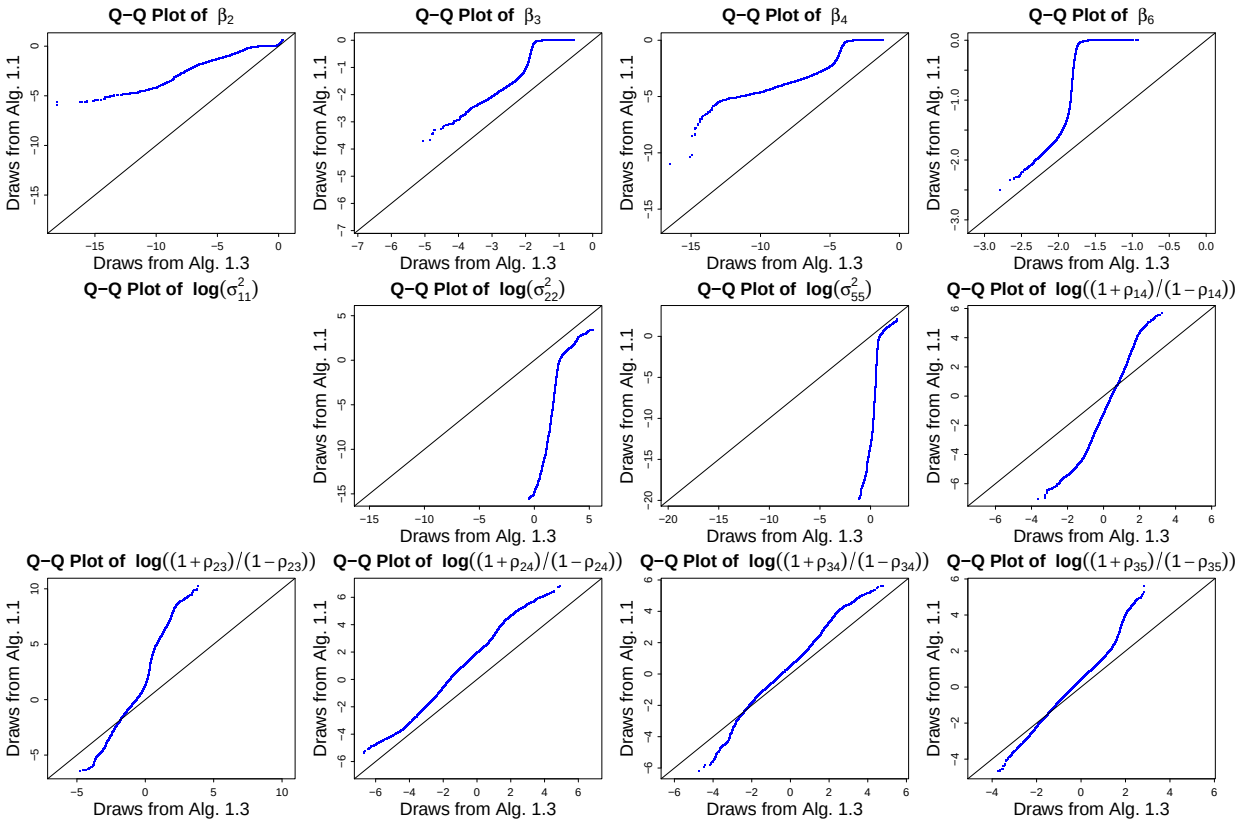

Figure 5: Quantile-quantile plots for comparing Algorithms 1.1 and 1.3 in the margarine-purchase data analysis.

5 Data Analysis

For a further comparison of the algorithms, we consider a data set describing margarine purchases which is available in the *bayesm* package of R. Following BN, we limit analysis to purchases of six brands: "Parkay stick", "Blue Bonnet stick", "Fleischmanns stick", "House brand stick", "Generic stick", and "Shedd Spread tub", and only consider the first purchase of one of these brands for each household. This results in a dataset consisting of $n = 507$ observations.

We set "Parkay stick" as the base category, and $p = 5$. Again following BN, we set up a model that only includes intercept terms for the other five categories and a coefficient for log prices. Thus $q = 6$, and $X_i = [I_p, g_i]$, where I_p is the identity matrix and g_i is the p-vector of differences in log prices between each category and the base. We again specify the prior distribution for Σ and α^2 as in Section 2, with $\nu = p$, $\alpha_0^2 = \nu$, and $S = \mathrm{Diag}(1, \ldots, 1)$, and for β, as $\beta \sim \mathrm{N}_q[0, \mathrm{Diag}(100, \ldots, 100)]$. When implementing Algorithms 1.1, 1.3, 2.1 and 2.2, we set the variance corresponding to "Blue Bonnet stick" as one. For each algorithm, we run a chain of length 300,000, discard the first 100,000 draws, and thin the rest draws by 10. In this way we obtain 20,000 draws from each algorithm.

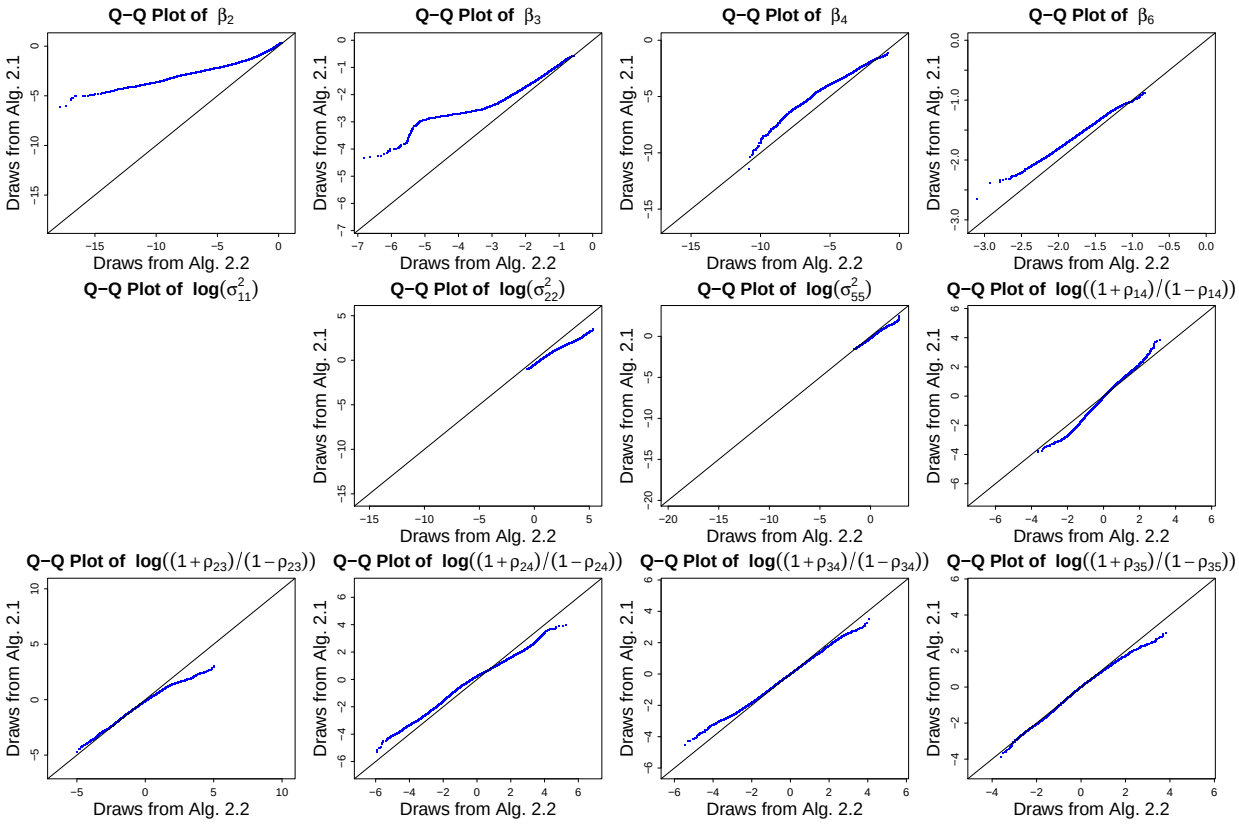

Figure 6: Quantile-quantile plots for comparing Algorithms 2.1 and 2.2 in the margarine-purchase data analysis.

Figures 5, 6, and 7 present the quantile-quantile plots of selected parameters correspondingly sampled with Algorithms 1.1 and 1.3, Algorithms 2.1 and 2.2, and Algorithms 3.1 and 3.2 respectively. The parameters we consider are β_2, β_3, β_4, β_6, $\log(\sigma_{11}^2)$, $\log(\sigma_{22}^2)$, $\log(\sigma_{55}^2)$, $\log\left(\frac{1+\rho_{14}}{1-\rho_{14}}\right)$, $\log\left(\frac{1+\rho_{23}}{1-\rho_{23}}\right)$, $\log\left(\frac{1+\rho_{24}}{1-\rho_{24}}\right)$, $\log\left(\frac{1+\rho_{34}}{1-\rho_{34}}\right)$, and $\log\left(\frac{1+\rho_{35}}{1-\rho_{35}}\right)$. They are selected because their stationary distributions show relatively obvious difference for all three pairs of original and corrected algorithms. We find that Algorithms 1.1, 2.1, and 3.1 all fail to deliver draws from the target posterior distribution. The situation is most substantial for Algorithm 1.1. Moreover, in terms of autocorrelation, Algorithm 1.3 performs substantially better than Algorithm 1.1, while Algorithms 2.2 and 3.2 perform similarly as Algorithms 2.1 and 3.1 respectively. In addition, Algorithms 1.3, 2.2, and 3.2 take around 15% more computational time than Algorithms 1.1, 2.1, and 3.1 respectively.

6 Conclusion

The algorithms of IvD and BN are implemented in the popular R package *MNP* and are widely used for fitting MNP models. We point out errors in these algorithms and propose corrections. Using both a

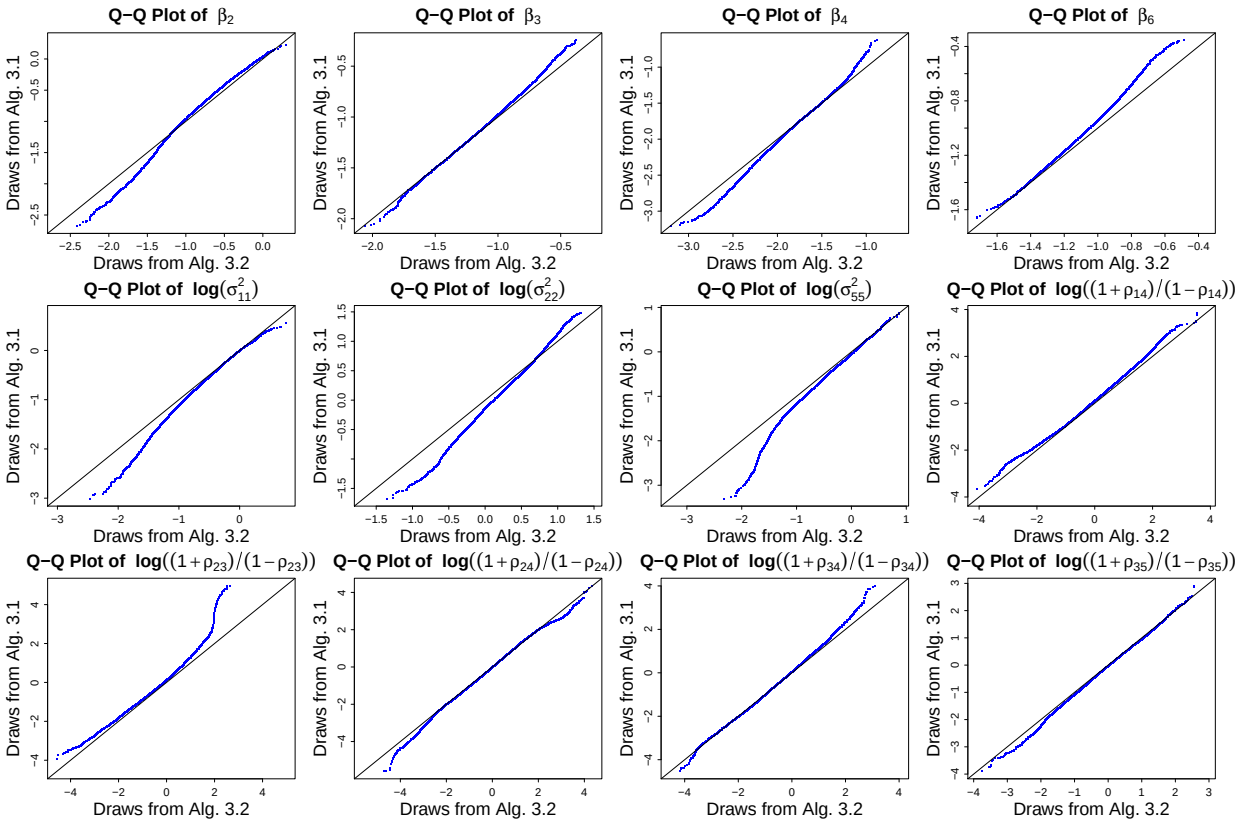

Figure 7: Quantile-quantile plots for comparing Algorithms 3.1 and 3.2 in the margarine-purchase data analysis.

simulation study and a real-data analysis, we illustrate the difference between the original and corrected algorithms. From these analyses, we find that the errors can significantly affect the final results, especially in that they alter the stationary distribution and hence the fitted parameters. Considering the popularity of these algorithms, it is important that they are corrected. We have done so here and will do it soon in the *MNP* package. The corrected algorithms require some what more computational time due to the additional rejection sampling steps, however, the extra computational time is small and at least in some cases it is made up by the improved autocorrelation of the corrected algorithms.

References

Berrett, C. and Calder, C. A. (2012). Data Augmentation Strategies for the Bayesian Spatial Probit Regression Model. *Computational Statistics and Data Analysis* **56**, 478–490.

Burgette, L. F. and Nordheim, E. V. (2012). The Trace Restriction: An Alternative Identification Strategy for the Bayesian Multinomial Probit Model. *Journal of Business and Economic Statistics* **30(3)**, 404–410.

Chaudoin, S. (2014). Audience Features and the Strategic Timing of Trade Disputes. *International Organization*

68(4), 877–911.

Hausman, J. and Wise, D. (1978). A Conditional Probit Model for Qualitative Choice: Discrete Decisions Recognizing Interdependence and Heterogeneous Preferences. *Econometrica* **45**, 319–339.

Horiuchi, Y., Imai, K., and Taniguchi, N. (2007). Designing and Analyzing Randomized Experiments: Application to a Japanese Election Survey Experiment. *American Journal of Political Science* **51(3)**, 669–687.

Hruschka, H. (2007). Using a Heterogeneous Multinomial Probit Model with a Neural Net Extension to Model Brand Choice. *Journal of Forecasting* **26**, 113–127.

Imai, K. and van Dyk, D. A. (2005a). A Bayesian Analysis of the Multinomial Probit Model Using Marginal Data Augmentation. *The Journal of Econometrics* **124**, 2, 311–334.

Imai, K. and van Dyk, D. A. (2005b). MNP: R Package for Fitting the Multinomial Probit Model. *Journal of Statistical Software* **14**, Issue 5, 1–32.

Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. M. (1998). Markov Chain Monte Carlo in Practice: A Roundtable Discussion. *American Statistician* **52**, 93–100.

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

Liu, J. S., Wong, W. H., and Kong, A. (1994). Covariance Structure of the Gibbs sampler with Applications to Comparisons of Estimators and Augmentation Schemes. *Biometrika* **81**, 27–40.

Lu, H., Tsai, F., Chen, H., Hung, M., and Li, S. (2012). Credit Rating Change Modeling Using News and Financial Ratios. *ACM Transactions on Management Information Systems* **3(3)**, Article No.14.

McCulloch, R., Polson, N., and Rossi, P. (2000). A Bayesian Analysis of the Multinomial Probit Model with Fully Identified Parameters. *Journal of Econometrics* **99**, 173–193.

McCulloch, R. E. and Rossi, P. E. (1994). An Exact Likelihood Analysis of the Multinomial Probit Model. *Journal of Econometrics* **64**, 207–240.

McFadden, D. (1989). A Method of Simulated Moments for Estimation of Discrete Response Models without Numerical Integration. *Econometrica* **57**, 995–1026.

Meng, X.-L. and van Dyk, D. A. (1999). Seeking Efficient Data Augmentation Schemes via Conditional and Marginal Augmentation. *Biometrika* **86**, 301–320.

Nobile, A. (1998). A Hybrid Markov Chain for the Bayesian Analysis of the Multinomial Probit Model. *Statistics and Computing* **8**, 229–242.

Nobile, A. (2000). Comment: Bayesian Multinomial Probit Models with Normalization Constraint. *Journal of Econometrics* **99**, 335–345.

Queralt, D. (2012). Economic Voting in Multi-Tiered Polities. *Electoral Studies* **31**, 107–119.

Robert, C. (1995). Simulation of Truncated Normal Variables. *Statistics and Computing* **5**, 121–125.

Sinclair, A. H. and Whitford, A. B. (2013). Separation and Integration in Public Health: Evidence from Organizational Structure in the States. *Journal of Public Administration Research and Theory* **23(1)**, 55–77.

Tanner, M. A. and Wong, W. H. (1987). The Calculation of Posterior Distributions by Data Augmentation (with

Discussion). *Journal of the American Statistical Association* **82**, 528–550.

van Dyk, D. A. and Meng, X.-L. (2001). The Art of Data Augmentation (with Discussion). *The Journal of Computational and Graphical Statistics* **10**, 1–111.

Vincent, T. L., Green, P. J., and Woolfson, D. N. (2013). LOGICOIL—Multi-State Prediction of Coiled-Coil Oligomeric State. *Bioinformatics* **29(1)**, 69–76.

Zhang, X., Boscardin, W. J., and Belin, T. R. (2008). Bayesian Analysis of Multivariate Nominal Measures Using Multivariate Multinomial Probit Models. *Computational Statistics and Data Analysis* **52**, 3697–3708.

Algorithm 2.1

Step 0: Initialize parameters $t = 0$, $\beta^{(0)}$, $\alpha^{(0)}$, $\Sigma^{(0)}$ and $W^{(0)}$.

while $t < T$ **do**

Step 1: Update $\left((\alpha^2)^\star, Z\right)$ from $p(\alpha^2, Z|Y, \beta^{(t)}, \Sigma^{(t)})$ by

(a) sampling $(\alpha^2)^\star$ from $p(\alpha^2|\Sigma^{(t)})$: $(\alpha^2)^\star \sim \alpha_0^2 \text{trace}\left(S{\Sigma^{(t)}}^{-1}\right)/\chi^2_{\nu p}$;

(b) sampling Z from $p(Z|Y, (\alpha^2)^\star, \beta^{(t)}, \Sigma^{(t)})$:

for $i := 1, \ldots, n$ **do**

for $k := 1, \ldots, p$ **do**

sampling $W^\star_{ik}$ via $p(W_{ik}|Y_i, W^\star_{i,-k}, \beta^{(t)}, \Sigma^{(t)})$: $W^\star_{ik} \sim \text{TN}(\mu_{ik}, \tau^2_{ik})$, see Appendix C for details;

end for

Set $Z_i = \alpha^\star(W^\star_i - X_i\beta^{(t)})$.

end for

Step 2: Update $\left((\alpha^2)^{(t+1)}, \Sigma^{(t+1)}\right)$ via $p(\alpha^2, \Sigma|Y, Z, \beta^{(t)})$ by

(a) sampling $\tilde{\Sigma}^\star$ from $p(\tilde{\Sigma}|Y, Z, \beta^{(t)})$:

$$\tilde{\Sigma}^\star \sim \text{Inv-Wishart}\left[n + \nu, \sum_{i=1}^n Z_i {Z_i}^{\text{T}}\right];$$

(b) setting $\alpha^{(t+1)} = \tilde{\sigma}^\star_{11}$, $\Sigma^{(t+1)} = \tilde{\Sigma}^\star/\left(\alpha^{(t+1)}\right)^2$, and $W_i^{(t+1)} = (Z_i + \alpha^{(t+1)}X_i\beta^{(t)})/\alpha^{(t+1)}$.

Step 3: Update $\beta^{(t+1)}$ via $p(\beta|Y, W^{(t+1)}, \Sigma^{(t+1)})$:

$$\beta^{(t+1)} \sim \text{N}_q\left[\hat{\beta}, \left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t+1)}}^{-1}X_i + A^{-1}\right)^{-1}\right],$$

where $\hat{\beta} = \left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t+1)}}^{-1}X_i + A^{-1}\right)^{-1}\left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t+1)}}^{-1}W_i^{(t+1)}\right)$.

return $\beta^{(t+1)}$, $\Sigma^{(t+1)}$, and $W^{(t+1)}$

$t + 1 \leftarrow t$

end while

APPENDIX: Details of Algorithms 2.1–3.2

A Algorithms 2.1 and 2.2

Algorithm 2 of IvD does not marginalize α out when updating β. We call this algorithm Algorithm 2.1 in this paper. Algorithm 2.1 can be used when the prior mean of β, β_0, is not equal to zero, while Algorithm 1.1 can not.

The error arises in Step 2(a), which is the same as the error in Step 3(a) of Algorithm 1.1. Thus the correction to Algorithm 2.1 is

Algorithm 2.2: Steps 0, 1, and 3 of Algorithm 2.2 are the same as Algorithm 2.1. In Step 2(a), however, Algorithm 2.2 updates $\tilde{\Sigma}^\star$ by sampling from a constrained inverse-Wishart distribution, that is,

$$\tilde{\Sigma}^\star \sim \text{Inv-Wishart}\left(n + \nu, \sum_{i=1}^n Z_i Z_i^{\text{T}}\right) \text{ subject to the constraint in (10).}$$

Algorithm 3.1

Step 0: Initialize parameters $t = 0$, $\beta^{(0)}$, $\alpha^{(0)}$, $\Sigma^{(0)}$ and $W^{(0)}$.

while $t < T$ **do**

Step 1: Update $\left((\alpha^2)^\star, \tilde{W}^\star\right)$ via $p(\alpha^2, \tilde{W}|Y, \beta^{(t)}, \Sigma^{(t)})$ by

(a) sampling $(\alpha^2)^\star$ from $p(\alpha^2|\Sigma^{(t)})$: $(\alpha^2)^\star \sim \alpha_0^2 \text{trace}\left(S{\Sigma^{(t)}}^{-1}\right)/\chi^2_{\nu p}$;

(b) sampling $\tilde{W}^\star$ from $p(\tilde{W}^\star|Y, (\alpha^2)^\star, \beta^{(t)}, \Sigma^{(t)})$:

for $i := 1, \ldots, n$ **do**

for $k := 1, \ldots, p$ **do**

sampling $W^\star_{ik}$ from $p(W_{ik}|Y_i, W^\star_{i,-k}, \beta^{(t)}, \Sigma^{(t)})$: $W^\star_{ik} \sim \text{TN}(\mu_{ik}, \tau^2_{ik})$, see Appendix C for details;

end for

Set $\tilde{W}^\star_i = \alpha^\star W^\star_i$.

end for

Step 2: Update $\left((\alpha^2)^\star, \beta^{(t+1)}\right)$ via $p(\alpha^2, \beta|Y, \tilde{W}^\star, \Sigma^{(t)})$ by

(a) sampling $(\alpha^2)^\star$ from $p(\alpha^2|Y, \tilde{W}^\star, \Sigma^{(t)})$:

$$(\alpha^2)^\star \sim \frac{\sum_{i=1}^n (\tilde{W}^\star_i - X_i\hat{\beta})^{\text{T}}{\Sigma^{(t)}}^{-1}(\tilde{W}^\star_i - X_i\hat{\beta}) + \hat{\beta}^{\text{T}}A^{-1}\hat{\beta} + \text{trace}\left(\tilde{S}{\Sigma^{(t)}}^{-1}\right)}{\chi^2_{(n+\nu)p}},$$

where $\hat{\beta} = \left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t)}}^{-1}X_i + A^{-1}\right)^{-1}\left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t)}}^{-1}\tilde{W}^\star_i\right)$;

(b) sampling $\tilde{\beta}^\star$ from $p(\tilde{\beta}|Y, \tilde{W}^\star, (\alpha^2)^\star, \Sigma^{(t)})$:

$$\tilde{\beta}^\star \sim \text{N}_q\left[\hat{\beta}, (\alpha^2)^\star\left(\sum_{i=1}^n X_i^{\text{T}}{\Sigma^{(t)}}^{-1}X_i + A^{-1}\right)^{-1}\right],$$

and setting $\beta^{(t+1)} = \tilde{\beta}^\star/\alpha^\star$.

Step 3: Update $\left((\alpha^2)^{(t+1)}, \Sigma^{(t+1)}\right)$ via $p(\alpha^2, \Sigma|Y, \tilde{W}^\star, \beta^{(t+1)})$ by

(a) sampling $\tilde{\Sigma}^\star$ from $p(\tilde{\Sigma}|Y, Z, \beta^{(t+1)})$:

$$\tilde{\Sigma}^\star \sim \text{Inv-Wishart}\left(n + \nu, \sum_{i=1}^n Z_i Z_i^{\text{T}}\right),$$

where $Z_i = \tilde{W}^\star_i - \alpha^\star X_i\beta^{(t+1)}$;

(b) setting $\alpha^{(t+1)} = \sqrt{\text{trace}(\tilde{\Sigma}^\star/p)}$, $\Sigma^{(t+1)} = \tilde{\Sigma}^\star/\left(\alpha^{(t+1)}\right)^2$, $\beta^{(t+1)} = \tilde{\beta}^\star/\alpha^{(t+1)}$, and $W^{(t+1)} = \tilde{W}^\star/\alpha^{(t+1)}$.

return $\beta^{(t+1)}$, $\Sigma^{(t+1)}$, and $W^{(t+1)}$

$t + 1 \leftarrow t$

end while

Note that $\beta^{(t+1)}$ in $\tilde{Z}_i\left(\tilde{\sigma}^\star_{11}\right)$ of the constraint (10) should be replaced by $\beta^{(t)}$ in Algorithm 2.2.

B Algorithms 3.1 and 3.2

We call the algorithm of BN Algorithm 3.1 in this paper. Algorithm 3.1 is almost the same as Algorithm 1.1. The only difference is Step 3(b). Specifically, first, in Algorithm 3.1, α^2 in this step is set to $\text{trace}(\tilde{\Sigma})/p$, while in Algorithm 1.1, α^2 is set to the first element of $\tilde{\Sigma}$; second, Algorithm 3.1 sets $\beta = \tilde{\beta}/\alpha$ in Step 3(b), while Algorithm 1.1 not.

Besides applying the two corrections to Step 3 of Algorithm 3.1, we further remove "$\beta^{(t+1)} = \tilde{\beta}^\star/\alpha^{(t+1)}$"

in Step 3(b) of Algorithm 3.1, because we update $\tilde{\Sigma}^\star$ conditioning on $(Y, Z, \beta^{(t+1)})$, not on $(Y, \tilde{W}^\star, \tilde{\beta}^\star)$. Thus, we get

Algorithm 3.2: This algorithm completely corrects Algorithm 3.1. In particular, Steps 0, 1, and 2 of Algorithm 3.2 are the same as Algorithm 3.1. In Step 3(a), however, Algorithm 3.2 updates $\tilde{\Sigma}^\star$ by sampling from a constrained inverse-Wishart distribution, that is,

$$\tilde{\Sigma}^\star \sim \text{Inv-Wishart}\left(n+\nu, \sum_{i=1}^{n} Z_i Z_i^{\mathrm{T}}\right) \text{ subject to the constraint } (10^\star).$$

Constraint $(10^\star)$ is an adaption of (10) by replacing $\tilde{\sigma}_{11}^\star$ with $r = \sqrt{\text{trace}(\tilde{\Sigma}^\star/p)}$. Specifically, $\tilde{Z}_i(r) = Z_i + rX_i\beta^{(t+1)}$, for $i = 1, \ldots, n$. The updated value of r must satisfy

$$\begin{cases} \max\left\{\tilde{Z}_{i1}(r), \ldots, \tilde{Z}_{ip}(r)\right\} < 0 & \text{if } Y_i = 0 \\ \max\left\{0, \tilde{Z}_{i1}(r), \ldots, \tilde{Z}_{ip}(r)\right\} = \tilde{Z}_{ik}(r) & \text{if } Y_i = k \end{cases}, \quad \text{for} \quad i = 1, \ldots, n. \tag{$10^\star$}$$

Finally, in Step 3(b), Algorithm 3.2 sets $\alpha^{(t+1)} = \sqrt{\text{trace}(\tilde{\Sigma}^\star/p)}$, $\Sigma^{(t+1)} = \tilde{\Sigma}^\star/\left(\alpha^{(t+1)}\right)^2$, and $W_i^{(t+1)} = (Z_i + \alpha^{(t+1)} X_i \beta^{(t+1)})/\alpha^{(t+1)}$.

C Details of Sampling W in Step 1(b) of Algorithms 1.1–3.2

Updating W in Step 1(b) of Algorithms 1.1–3.2 consists of sampling from a series of univariate truncated normal distributions, that is, for $i = 1, \ldots, n$ and $k = 1, \ldots, p$,

$$W_{ik}^\star \sim \text{TN}(\mu_{ik}, \tau_{ik}^2),$$

where $\mu_{ik} = X_{ik}\beta^{(t)} + \Sigma_{k,-k}^{(t)} \Sigma_{-k,-k}^{(t)-1} (W_{i,-k}^\star - X_{i,-k}\beta^{(t)})$ with $W_{i,-k}^\star = \left(W_{i1}^\star, \ldots, W_{i,(k-1)}^\star, W_{i,(k+1)}^{(t)}, \ldots, W_{ip}^{(t)}\right)$, and $\tau_{ik}^2 = \left(\sigma_{kk}^{(t)}\right)^2 - \Sigma_{k,-k}^{(t)} \Sigma_{-k,-k}^{(t)-1} \Sigma_{-k,k}^{(t)}$; X_{ik} is the kth row of X_i, and $X_{i,-k}$ is the sub-matrix of X_i with X_{ik} removed. The constraint on $W_{ik}^\star$ is, $W_{ik}^\star \geq \max\{0, W_{i,-k}^\star\}$, if $Y_i = k$; $W_{ik}^\star < 0$, if $Y_i = 0$; and $W_{ik}^\star \leq \max\{0, W_{ij}^\star\}$, if $Y_i = j \neq k$.

If the constraint on $W_{ik}^\star$ has the form, $W_{ik}^\star \geq w$, and $w \leq 0$, we update $W_{ik}^\star$ with simple rejection sampling: we iteratively sample from the unconstrained normal distribution until $W_{ik}^\star \geq w$ is satisfied. If $W_{ik}^\star \geq w$, but $w > 0$, we update $W_{ik}^\star$ with the exponential rejection sampling proposed by Robert (1995). If the constraint on $W_{ik}^\star$ has the form $W_{ik}^\star \leq w$, we can apply the above sampling scheme with slight adaption, since $-W_{ik}^\star \geq -w$.