Journal of Modern Applied Statistical Methods
November 2009, Vol. 8, No. 2, 646-658

1538 – 9472/09/$95.00

JMASM ALGORITHMS & CODE

JMASM28: Gibbs Sampling for 2PNO Multi-unidimensional Item Response Theory Models (Fortran)

Yanyan Sheng Todd C. Headrick
Southern Illinois University Carbondale

A Fortran 77 subroutine is provided for implementing the Gibbs sampling procedure to a multi-unidimensional IRT model for binary item response data with the choice of uniform and normal prior distributions for item parameters. In addition to posterior estimates of the model parameters and their Monte Carlo standard errors, the algorithm also estimates the correlations between distinct latent traits. The subroutine requires the user to have access to the IMSL library. The source code is available at http://www.siuc.edu/~epse1/sheng/Fortran/MUIRT/GSMU2.FOR. An executable file is also provided for download at http://www.siuc.edu/~epse1/sheng/Fortran/MUIRT/EXAMPLE.zip to demonstrate the implementation of the algorithm on simulated data.

Key words: multi-unidimensional IRT model, two-parameter normal ogive model, MCMC, Gibbs sampling, Fortran.

Introduction

Modeling the interaction of a person's trait and the test at the item level for binary response data, the conventional item response theory (IRT) models rely on a strong assumption of unidimensionality. That is, each test item is designed to measure some facet of a unified latent trait. However, psychological processes have consistently been found to be more complex and an increasing number of educational measurements assess a person on more than one latent trait. In the situations when a test consists of several subtests with each measuring one latent trait, the multi-unidimensional IRT models have been found to be more appropriate than the unidimensional models (Sheng & Wikle, 2007), as they allow inferences to be made about a person for each distinct trait being measured.

For the two-parameter normal ogive (2PNO) multi-unidimensional model, the probability of person i obtaining a correct response for item j in subtest v, where $i = 1,...,n$, $j = 1,...,k_v$, $v = 1,...,m$, and $K = \sum_v k_v$, is defined as

$$P(y_{vij} = 1) = \Phi(\alpha_{vj}\theta_{vi} - \gamma_{vj}) = \int_{-\infty}^{\alpha_{vj}\theta_{vi}-\gamma_{vj}} \frac{1}{\sqrt{2\pi}} e^{\frac{-t^2}{2}} dt \quad (1)$$

(e.g., Lee, 1995; Sheng & Wikle, 2007), where α_{vj} and θ_{vi} are scalar parameters representing the item discrimination and the continuous person trait in the vth latent dimension, and γ_{vj} is a scalar parameter indicating the location in that dimension where the item provides maximum information. To estimate both item and person parameters simultaneously, Markov

Yanyan Sheng is an Associate Professor of Measurement and Statistics. Her areas of research interest are psychometrics, IRT and Bayesian hierarchical models, adaptive testing. Email: ysheng@siu.edu. Todd C. Headrick is Professor of Statistics. Address: Section on Statistics and Measurement, Department of EPSE, 222-J Wham Building, Mail Code 4618, Southern Illinois University-Carbondale, IL, 62901. His areas of research interest are statistical computing, nonparametric statistics, and optimization. Email: headrick@siu.edu.

chain Monte Carlo (MCMC; e.g., Chib & Greenberg, 1995) techniques are used to summarize the posterior distributions that arise in the context of the Bayesian prior-posterior framework (Carlin & Louis, 2000; Chib & Greenberg, 1995; Gelfand & Smith, 1990; Gelman, Carlin, Stern, & Rubin, 2003; Tanner & Wong, 1987). Lee (1995) applied Gibbs sampling (Casella & George, 1992; Gelfand & Smith, 1990; Geman & Geman, 1984), an MCMC algorithm, to the 2PNO multi-unidimensional model and illustrated the model parameterization by adopting non-informative priors for item parameters.

Due to the reasons that informative priors are desirable in some applications in the Bayesian framework, and MCMC is computational demanding (see Sheng & Headrick, 2007, for a description of the problems), this study focuses on using Fortran, the fastest programming language for numerical computing (Brainerd, 2003) to implement the procedure. In particular, the paper provides a Fortran subroutine that obtains the posterior estimates and Monte Carlo standard errors of estimates for the item and person parameters in the 2PNO multi-unidimensional IRT model, as well as the posterior estimates of the correlations between the distinct latent traits. The subroutine allows the user to specify non-informative and informative priors for item parameters.

Methodology

The Gibbs Sampling Procedure

To implement Gibbs sampling to the 2PNO multi-unidimensional IRT model defined in (1), a latent continuous random variable Z is introduced so that $Z_{vij} \sim N(\alpha_{vj}\theta_{vi} - \gamma_{vj}, 1)$ (Albert, 1992; Lee, 1995; Tanner & Wong, 1987). Next, denote each person's latent traits for all items as $\boldsymbol{\theta}_i = (\theta_{1i}, ..., \theta_{mi})'$, which is assumed to have a multivariate normal (MVN) distribution, $\boldsymbol{\theta}_i \sim N_m(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a correlation matrix, and ρ_{st} is the correlation between θ_{si} and θ_{ti}, $s \neq t$, on the off diagonals. It may be noted that the unidimensional IRT model is a special case of the multi-unidimensional model where $\rho_{st} = 1$ for all s, t. Then, an unconstrained covariance matrix $\boldsymbol{\Sigma}^*$ is introduced (Lee, 1995), where $\boldsymbol{\Sigma}^* = [\sigma_{ij}]_{m \times m}$, so that the correlation matrix $\boldsymbol{\Sigma}$ can be easily transformed from $\boldsymbol{\Sigma}^*$ using $\rho_{st} = \frac{\sigma_{st}}{\sigma_s \sigma_t} (s \neq t)$. A non-informative prior is assumed for $\boldsymbol{\Sigma}^*$ so that $p(\Sigma^*) \propto |\Sigma^*|^{-\frac{m+1}{2}}$. Hence, with prior distributions assumed for ξ_{vj}, where $\xi_{vj} = (\alpha_{vj}, \gamma_{vj})'$, the joint posterior distribution for $(\boldsymbol{\theta}, \xi, \mathbf{Z}, \boldsymbol{\Sigma}^*)$ is

$$p(\boldsymbol{\theta}, \xi, \mathbf{Z}, \boldsymbol{\Sigma}^* | \mathbf{y}) \propto f(\mathbf{y} | \mathbf{Z}) p(\mathbf{Z} | \boldsymbol{\theta}, \xi) p(\xi) p(\boldsymbol{\theta} | \boldsymbol{\Sigma}) p(\boldsymbol{\Sigma}^*). \quad (2)$$

where $f(\mathbf{y} | \mathbf{Z})$ is the likelihood function.

With non-informative priors for α_{vj} and γ_{vj} so that $\alpha_{vj} > 0$ and $p(\gamma_{vj}) \propto 1$, the full conditional distributions of Z_{vij}, $\boldsymbol{\theta}_i$, ξ_{vj} and $\boldsymbol{\Sigma}^*$ can be derived in closed forms as follows:

$$Z_{vij} | \bullet \sim \begin{cases} N_{(0,\infty)}(\alpha_{vj}\theta_{vi} - \gamma_{vj}, 1), \ if \ y_{vij} = 1 \\ N_{(-\infty,0)}(\alpha_{vj}\theta_{vi} - \gamma_{vj}, 1), \ if \ y_{vij} = 0 \end{cases}; \quad (3)$$

$$\boldsymbol{\theta}_i | \bullet \sim N_m((\mathbf{A}'\mathbf{A} + \boldsymbol{\Sigma})^{-1}\mathbf{A}'\mathbf{B}, (\mathbf{A}'\mathbf{A} + \boldsymbol{\Sigma})^{-1}), \quad (4)$$

where $\mathbf{A}_{(K \times m)} = \begin{pmatrix} \boldsymbol{\alpha}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\alpha}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\alpha}_m \end{pmatrix}$ and

$\mathbf{B}_{(K \times 1)} = \begin{pmatrix} \mathbf{Z}_{1i} + \boldsymbol{\gamma}_1 \\ \mathbf{Z}_{2i} + \boldsymbol{\gamma}_2 \\ \vdots \\ \mathbf{Z}_{mi} + \boldsymbol{\gamma}_m \end{pmatrix}$, $\boldsymbol{\alpha}_v = (\alpha_{v1}, ..., \alpha_{v,k_v})'$,

$\mathbf{Z}_{vi} = (Z_{vi1}, ..., Z_{nik_v})'$, $\boldsymbol{\gamma}_v = (\gamma_{v1}, ..., \gamma_{v,k_v})'$;

$$\xi_{vj} | \bullet \sim N_2((\mathbf{x}_v'\mathbf{x}_v)^{-1}\mathbf{x}_v'\mathbf{Z}_{vj}, (\mathbf{x}_v'\mathbf{x}_v)^{-1}) I(\alpha_{vj} > 0), \quad (5)$$

where $\mathbf{x}_v = [\theta_v, -1]$;

$$\boldsymbol{\Sigma}^* \mid \bullet \sim W^{-1}(\mathbf{S}^{-1}, n) \qquad (6)$$

(an inverse Wishart distribution), where $\mathbf{S} = \sum_{i=1}^{n} (c\boldsymbol{\theta}_i)(c\boldsymbol{\theta}_i)'$ and

$$c = \begin{pmatrix} (\prod_j \alpha_{1j})^{\frac{1}{k_1}} & 0 & \cdots & 0 \\ 0 & (\prod_j \alpha_{2j})^{\frac{1}{k_2}} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & (\prod_j \alpha_{mj})^{\frac{1}{k_m}} \end{pmatrix}$$

(see Lee, 1995 for a detailed derivation).

Alternatively, conjugate priors can be assumed for α_{vj} and γ_{vj} so that $\alpha_{vj} \sim N_{(0,\infty)}(\mu_{\alpha_v}, \sigma^2_{\alpha_v})$, $\gamma_{vj} \sim N(\mu_{\gamma_v}, \sigma^2_{\gamma_v})$. In this case, the full conditional distribution of ξ_{vj} is derived to be

$$\xi_{vj} \mid \bullet \sim N_2((\mathbf{x}_v'\mathbf{x}_v + \boldsymbol{\Sigma}^{-1}_{\xi_v})^{-1}(\mathbf{x}_v'\mathbf{Z}_{vj} + \boldsymbol{\Sigma}^{-1}_{\xi_v}\boldsymbol{\mu}_{\xi_v}), (\mathbf{x}'\mathbf{x} + \boldsymbol{\Sigma}^{-1}_{\xi_v})^{-1}) I(\alpha_{vj} > 0) \qquad (7)$$

where $\boldsymbol{\mu}_{\xi_v} = (\mu_{\alpha_v}, \mu_{\gamma_v})'$ and $\boldsymbol{\Sigma}_{\xi_v} = \begin{pmatrix} \sigma^2_{\alpha_v} & 0 \\ 0 & \sigma^2_{\gamma_v} \end{pmatrix}$.

Hence, with starting values $\boldsymbol{\theta}^{(0)}$, $\xi^{(0)}$, and $\boldsymbol{\Sigma}^{(0)}$, observations ($\mathbf{Z}^{(l)}$, $\boldsymbol{\theta}^{(l)}$, $\xi^{(l)}$, $\boldsymbol{\Sigma}^{(l)}$) can be simulated using the Gibbs sampler by iteratively drawing from their respective full conditional distributions specified in (3), (4), (5), and (6) (or (3), (4), (7), and (6)). In particular, to go from ($\mathbf{Z}^{(l-1)}$, $\boldsymbol{\theta}^{(l-1)}$, $\xi^{(l-1)}$, $\boldsymbol{\Sigma}^{(l-1)}$) to ($\mathbf{Z}^{(l)}$, $\boldsymbol{\theta}^{(l)}$, $\xi^{(l)}$, $\boldsymbol{\Sigma}^{(l)}$), it takes four transition steps:

1. Draw $\mathbf{Z}^{(l)} \sim p(\mathbf{Z} \mid \mathbf{y}, \boldsymbol{\theta}^{(l-1)}, \xi^{(l-1)})$;
2. Draw $\boldsymbol{\theta}^{(l)} \sim p(\boldsymbol{\theta} \mid \mathbf{Z}^{(l)}, \xi^{(l-1)}, \boldsymbol{\Sigma}^{(l-1)})$;
3. Draw $\xi^{(l)} \sim p(\xi \mid \mathbf{Z}^{(l)}, \boldsymbol{\theta}^{(l)})$;
4. Draw $\boldsymbol{\Sigma}^{*(l)} \sim p(\boldsymbol{\Sigma}^* \mid \boldsymbol{\theta}^{(l)}, \xi^{(l)})$, and transform $\boldsymbol{\Sigma}^{*(l)}$ to $\boldsymbol{\Sigma}^{(l)}$.

This iterative procedure produces a sequence of samples for the model parameters ($\boldsymbol{\theta}^{(l)}, \xi^{(l)}$) and the hyperparameter $\boldsymbol{\Sigma}^{(l)}$, $l = 0, \ldots, L$. To reduce the effect of the starting values, early iterations in the Markov chain are set as burn-ins to be discarded. Samples from the remaining iterations are then used to summarize the posterior density of item parameters ξ, distinct person trait parameters $\boldsymbol{\theta}$, and the correlation matrix $\boldsymbol{\Sigma}$. As with standard Monte Carlo, the posterior means of all the samples collected after burn-in are considered as estimates of the true parameters ξ, $\boldsymbol{\theta}$, and $\boldsymbol{\Sigma}$.

However, the Monte Carlo standard errors cannot be calculated using the sample standard deviations because subsequent samples in each Markov chain are autocorrelated (e.g., Patz & Junker, 1999). One approach to calculating them is through batching (Ripley, 1987). That is, with a long chain of samples being separated into contiguous batches of equal length, the Monte Carlo standard error for each parameter is then estimated to be the standard deviation of these batch means. The Monte Carlo standard error of estimate is hence a ratio of the Monte Carlo standard error and the square root of the number of batches.

The Fortran Subroutine

The subroutine initially sets the starting values for the model parameters, $\boldsymbol{\theta}$, ξ, and the hyperparameter $\boldsymbol{\Sigma}$, so that $\theta^{(0)}_{vi} = 0$, $\alpha^{(0)}_{vi} = 2$, $\gamma^{(0)}_{vi} = -\Phi^{-1}(\sum_i y_{vij}/n)\sqrt{5}$ (Albert, 1992), and $\boldsymbol{\Sigma}^{(0)} = \mathbf{I}$, with $\mathbf{I}$ being the identity matrix. It then iteratively draws random samples for $\mathbf{Z}$, $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}^*$ from their respective full conditional distributions specified in (3), (4) and (6). Samples for ξ_{vj} are simulated either from (5), where uniform priors are assumed for ξ_{vj}, or from (7), where normal priors are adopted with

$\mu_{\alpha_v} = \mu_{\gamma_v} = 0$ and $\sigma^2_{\alpha_v} = \sigma^2_{\gamma_v} = 1$. The algorithm continues until all the L samples are simulated. It then discards the early burn-in samples, and computes the posterior estimates and Monte Carlo standard errors of estimates for the model parameters, $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$, as well as the hyperparameter $\boldsymbol{\Sigma}$, using batching.

For example, consider binary responses of 2,000 persons to a total of 16 test items, which are further divided into two subtests so that the first half measures one latent trait and the second half measures another (i.e., n = 2,000, m = 2, k_1 = 8, k_2 = 8, and K = 16). Three dichotomous (0-1) data matrices were simulated from the item parameters shown in the first column of Tables 1 and 2, so the actual correlation (ρ) between the two distinct latent traits (θ_1, θ_2) was set to be 0.2, 0.5 and 0.8, respectively. The Gibbs sampler was implemented to each data set so that 10,000 samples were simulated with the first 5,000 taken to be burn-in. The remaining 5,000 samples were separated into 5 batches, each with 1,000 samples.

With the uniform or the normal prior distributions described previously, two sets of the posterior means for α_v, γ_v, and ρ as well as their Monte Carlo standard errors were obtained for each simulated data and are displayed in the rest of the tables. Note that in all the three simulated situations, item parameters were estimated with enough accuracy and the two sets of posterior estimates differed only in the third decimal place, signifying that the results are not sensitive to the choice of prior distributions for ξ_{vj}. In addition, the small values of the Monte Carlo standard errors of estimates suggested that the Markov chains with a run length of 10,000 and a burn-in period of 5,000 reached the stationary distribution. Further, note that the procedure recovered the latent structure accurately as well, as the posterior estimates of the correlation between the two distinct latent traits, displayed in the last row of Table 2, was close to the actual correlation in all the three situations. For this example where 2,000-by-16 data matrices were considered, each implementation took less than 25 minutes. The length of the chains may be increased to be as long as 50,000, which takes about 90-120 minutes for each execution.

Conclusion

This Fortran subroutine allows the user to choose between uniform and normal priors for the item parameters, α_v and γ_v. In addition, the user can modify the source code by assigning other values to μ_α , σ^2_α, and μ_γ, σ^2_γ to reflect different prior beliefs on their distributions. Convergence can be assessed by inspecting Monte Carlo standard errors, as well as by comparing the marginal posterior mean and standard deviation of each parameter computed for every 1,000 samples after the burn-ins. For the latter, identical values provide a rough indication of similar marginal posterior densities, which further indicates possible convergence of the Markov chain (Gelfand, Hills, Racine-Poon & Smith, 1990; Hoijtink & Molenaar, 1997).

Note that the algorithm adopts a correlation matrix in the prior distribution, $\boldsymbol{\theta}_i \sim N_m(\mathbf{0}, \boldsymbol{\Sigma})$, to solve the problem of model nonidentifiability (see e.g., Lee, 1995, for a description of the problem). Bafummi, Gelman, Park, and Kaplan (2005) provides an alternative solution to the problem.

References

Albert, J. H. (1992). Bayesian estimation of normal ogive item response curves using Gibbs sampling. *Journal of Educational Statistics*, *17*, 251-269.

Bafumi, J., Gelman, A., Park, D. K., & Kaplan, N. (2005). Practical issues in implementing and understanding Bayesian ideal point estimation. *Political Analysis*, *13*, 171-187.

Brainerd, W. (2003). The importance of Fortran in the 21st century. *Journal of Modern Statistical Methods*, *2*, 14-15.

Casella, G., & George, E. I. (1992). Explaining the Gibbs sampler. *The American Statistician*, *46*, 167-174.

Carlin, B. P., & Louis, T. A. (2000). *Bayes and empirical Bayes methods for data analysis* (*2nd Ed.*). London: Chapman & Hall.

Table 1: Posterior Estimates and Monte Carlo Standard Errors of Estimates (MCSEs) for α_v with Uniform and Normal Priors

	$\rho = .2$		$\rho = .5$		$\rho = .8$	
	Uniform	Normal	Uniform	Normal	Uniform	Normal
Parameters	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)
α_1						
0.0966	0.0838 (.0013)	0.0828 (.0011)	0.0869 (.0007)	0.0846 (.0012)	0.0830 (.0013)	0.0847 (.0007)
0.0971	0.0675 (.0010)	0.0660 (.0013)	0.0731 (.0008)	0.0740 (.0010)	0.0657 (.0014)	0.0689 (.0012)
0.4589	0.4698 (.0035)	0.4704 (.0026)	0.4748 (.0028)	0.4707 (.0021)	0.4829 (.0021)	0.4797 (.0021)
0.9532	0.8556 (.0039)	0.8531 (.0069)	0.8804 (.0054)	0.8753 (.0058)	0.8937 (.0063)	0.8928 (.0045)
0.0771	0.0510 (.0009)	0.0502 (.0005)	0.0552 (.0013)	0.0550 (.0008)	0.0589 (.0007)	0.0577 (.0008)
0.4891	0.4900 (.0020)	0.4895 (.0024)	0.4855 (.0029)	0.4864 (.0012)	0.4659 (.0017)	0.4649 (.0017)
0.8599	1.0401 (.0185)	1.0348 (.0114)	1.0180 (.0080)	1.0120 (.0069)	0.9983 (.0057)	0.9930 (.0061)
0.9427	0.9381 (.0075)	0.9327 (.0024)	0.9477 (.0085)	0.9408 (.0088)	0.9628 (.0033)	0.9479 (.0075)
α_2						
0.2727	0.3013 (.0010)	0.2973 (.0026)	0.2654 (.0006)	0.2685 (.0014)	0.2348 (.0016)	0.2358 (.0013)
0.6532	0.7279 (.0051)	0.7251 (.0061)	0.6354 (.0028)	0.6346 (.0020)	0.7188 (.0042)	0.7142 (.0028)
0.1002	0.1231 (.0010)	0.1226 (.0014)	0.1528 (.0008)	0.1527 (.0012)	0.1088 (.0012)	0.1108 (.0018)
0.2339	0.0945 (.0014)	0.0965 (.0026)	0.1557 (.0021)	0.1535 (.0015)	0.1683 (.0020)	0.1670 (.0013)
0.9291	0.8554 (.0155)	0.8552 (.0131)	0.8145 (.0042)	0.8184 (.0071)	0.9208 (.0039)	0.9149 (.0061)
0.8618	0.8730 (.0128)	0.8575 (.0095)	0.9107 (.0060)	0.9001 (.0069)	0.9067 (.0034)	0.9055 (.0050)
0.0908	0.0543 (.0006)	0.0518 (.0016)	0.0556 (.0005)	0.0570 (.0007)	0.0463 (.0010)	0.0464 (.0007)
0.2083	0.2003 (.0006)	0.1967 (.0021)	0.2045 (.0016)	0.2035 (.0010)	0.2339 (.0013)	0.2351 (.0007)

Table 2: Posterior Estimates and Monte Carlo Standard Errors of Estimates (MCSEs) for γ_v and ρ with Uniform and Normal Priors

	$\rho = .2$		$\rho = .5$		$\rho = .8$	
	Uniform	Normal	Uniform	Normal	Uniform	Normal
Parameters	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)	Estimate (MCSE)
γ_1						
0.3629	0.3457 (.0007)	0.3447 (.0003)	0.3467 (.0010)	0.3448 (.0005)	0.3450 (.0004)	0.3452 (.0005)
-0.9010	-0.8881 (.0003)	-0.8875 (.0002)	-0.8891 (.0006)	-0.8885 (.0006)	-0.8875 (.0005)	-0.8865 (.0003)
-0.9339	-0.9288 (.0017)	-0.9277 (.0017)	-0.9288 (.0018)	-0.9270 (.0012)	-0.9317 (.0015)	-0.9310 (.0011)
-0.3978	-0.3976 (.0023)	-0.3983 (.0017)	-0.4035 (.0018)	-0.4012 (.0016)	-0.4059 (.0020)	-0.4062 (.0018)
0.3987	0.4077 (.0003)	0.4076 (.0008)	0.4085 (.0006)	0.4072 (.0006)	0.4073 (.0002)	0.4066 (.0007)
0.1654	0.1679 (.0003)	0.1681 (.0005)	0.1675 (.0009)	0.1666 (.0007)	0.1665 (.0010)	0.1669 (.0008)
-0.8108	-0.8302 (.0082)	-0.8294 (.0062)	-0.8232 (.0032)	-0.8186 (.0039)	-0.8122 (.0030)	-0.8091 (.0030)
-0.8012	-0.7091 (.0025)	-0.7064 (.0019)	-0.7145 (.0043)	-0.7102 (.0043)	-0.7186 (.0012)	-0.7140 (.0048)
γ_2						
0.2452	0.2902 (.0008)	0.2896 (.0007)	0.3122 (.0005)	0.3109 (.0002)	0.3037 (.0006)	0.3047 (.0005)
0.9792	1.0954 (.0031)	1.0941 (.0032)	1.0476 (.0015)	1.0461 (.0024)	1.1095 (.0021)	1.1045 (.0016)
-0.0190	-0.0216 (.0006)	-0.0212 (.0005)	-0.0058 (.0006)	-0.0068 (.0002)	-0.0200 (.0005)	-0.0196 (.0009)
0.8749	0.9549 (.0005)	0.9536 (.0006)	0.9624 (.0008)	0.9616 (.0009)	0.9568 (.0014)	0.9538 (.0005)
-0.3119	-0.2139 (.0026)	-0.2143 (.0013)	-0.2049 (.0019)	-0.2068 (.0011)	-0.2250 (.0005)	-0.2256 (.0011)
0.2005	0.2902 (.0025)	0.2888 (.0024)	0.2781 (.0021)	0.2735 (.0019)	0.2777 (.0012)	0.2750 (.0022)
0.4626	0.4658 (.0011)	0.4638 (.0004)	0.4514 (.0004)	0.4501 (.0002)	0.4550 (.0005)	0.4545 (.0012)
0.7184	0.7528 (.0008)	0.7510 (.0007)	0.7485 (.0007)	0.7462 (.0007)	0.7738 (.0003)	0.7723 (.0013)
ρ	0.1850 (.0022)	0.1853 (.0018)	0.5209 (.0031)	0.5213 (.0036)	0.7872 (.0037)	0.7942 (.0041)

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician, 49*, 327-335.

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association, 85*, 398-409.

Gelfand, A. E., Hills, S. E., Racine-Poon, A., & Smith, A. F. M. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association, 85*, 315-331.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence, 6*, 721-741.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. Boca Raton: Chapman & Hall/CRC.

Hoijtink, H., & Molenaar, I. W. (1997). A multidimensional item response model: Constrained latent class analysis using posterior predictive checks. *Psychometrika, 62*, 171-189.

Lee, H. (1995). *Markov chain Monte Carlo methods for estimating multidimensional ability in item response analysis.* Ph.D. Dissertation, University of Missouri, Columbia, MO.

Patz, R. J., & Junker, B. W. (1999). A straightforward approach to Markov chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics, 24*, 146-178.

Ripley, B. D. (1987). *Stochastic simulation*. NY: Wiley.

Sheng, Y., & Headrick, T. C. (2007). An algorithm for implementing Gibbs sampling for 2PNO IRT models. *Journal of Modern Applied Statistical Methods, 6*, 341-249.

Sheng, Y., & Wikle, C. K. (2007). Comparing multi-unidimensional and unidimensional IRT Models. *Educational & Psychological Measurement, 67*, 899-919.

Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distribution by data augmentation (with discussion). *Journal of the American Statistical Association, 82*, 528-550.

Appendix: Fortran Subroutine

```
      SUBROUTINE GSMU2(Y, N, K, M, MN, L, BURNIN, BN, UNIF, ITEM, PER, RPER)
C*************************************************************************
C Y = the n-by-K binary item response data
C N = the number of subjects
C K = the test length (total number of items)
C M = the number of subtests
C MN = an array with numbers of items in the m subtests
C L = the number of iterations using Gibbs sampling
C BURNIN = the early number of iterations that are to be discarded
C BN  = the number of batches
C UNIF = a 0-1 indicator with 0 specifying normal priors for item
C     parameters and 1 specifying uniform priors for them
C ITEM = a K-by-4 matrix of posterior estimates and MCSEs for item
C     parameters
C PER = a n-by-2m matrix of posterior estimates and MCSEs for person
C      traits
C RPER = a (m*(m-1)/2)-by-2 matrix of posterior estimates and MCSEs
C     for the correlation(s) between person traits
C*************************************************************************
      INTEGER       N, K, MN(M), L, Y(N,K), IRANK, INDX(M), UNIF, COUNT,
     &              BURNIN, BSIZE, BN
      REAL    A(K), G(K), TH(N,M), AA(K,M), ZLP(N,K), LP, Z(N,K), PHAT(K),
     &       U, PVAR(M, M), SIGMA(M,M), RSIG(M,M), PVAR1(M,M), RTH(M),
```

```
     &       BA(K), PMEAN1(M), PMEAN(M), X(N,2), XX(2,2), IX(2,2), RMN(M),
     &       ZV(N,1), XZ(2,1), AMAT(2, 2), BZ(2,1), AMU, GMU, AVAR, GVAR,
     &       AGMU(2,1), AGVAR(2,2), AGSIG(2,2), BETA(2), C(M,M), CTH(M,N),
     &       D(M,M), AV(L,K), GV(L,K), RHO(M,M,L), THV(N,M,L), ITEM(K,4),
     &       PER(N,2*M), SUM0, SUM1, SUM2, SUM3,  M0, M1, M2, M3, TOT,
     &       TOT1, TOT2, TOT3, SS, SS1, SS2, SS3, RPER(M*(M-1)/2,2),
     &       PRODA, VAR(M,M)
      DOUBLE PRECISION  BB, TMP
C*************************************************************************
C Connect to external libraries for normal (RNNOR), uniform (RNUN), and
C multivariate normal (RNMVN) random number generator, inverse (DNORIN)
C and CDF (ANORDF, DNORDF) for the standard normal distribution, and
C Cholesky factorization (CHFAC) routines
C*************************************************************************
      EXTERNAL    RNNOR, RNUN, ANORDF, CHFAC, DNORDF, DNORIN, RNMVN
C*************************************************************************
C Set initial values for item parameters a(v), g(v), person ability
C theta, and the hyperparameter sigma, so a(v)=2, g(v)=-Φ^-1(Σ_i y_ij / n)√5
C for all k(v) items, theta(v)=0 for all n person traits, and sigma=I
C*************************************************************************
      PHAT = SUM(Y, DIM = 1)
      DO 10 I = 1, K
            A(I) = 2.0
            G(I) = -ANORIN(PHAT(I)/FLOAT(N))*SQRT(5.0)
   10 CONTINUE
      DO 15 I = 1, N
      DO 15 J = 1, M
            TH(I, J) = 0.0
   15 CONTINUE
      DO 20 I=1,M
      DO 20 J=1,M
            SIGMA(I, J) = 0.0
            SIGMA(I, I) = 1.0
   20 CONTINUE
      RMN = FLOAT(MN)
C*************************************************************************
C Start iteration
C*************************************************************************
      COUNT = 0
      DO 30 IT = 1, L
            COUNT = COUNT + 1
            DO 40 I = 1, K
            DO 40 J = 1, M
                  AA(I, J) = 0.0
   40       CONTINUE
            JJ = 0
            DO 50 I = 1, M
                  J = 1
                  DO WHILE (J .LE. MN(I))
                        JJ = JJ+1
                        AA(JJ, I) = A(JJ)
                        J = J + 1
                  END DO
   50       CONTINUE
```

```
C*************************************************************************
C Update samples for Z from its normal posterior distributions
C*************************************************************************
             ZLP = MATMUL(TH, TRANSPOSE(AA))
             DO 60 I = 1, N
             DO 60 J = 1, K
                   LP = ZLP(I, J) - G(J)
                   BB = ANORDF(0.0 - LP)
                   CALL RNUN(1, U)
                   TMP = BB*(1-Y(I, J)) + (1-BB)*Y(I, J))*U + BB*Y(I, J)
                   Z(I, J) = DNORIN(TMP) + LP
   60        CONTINUE
C*************************************************************************
C Update samples for theta from their MVN posterior distributions
C*************************************************************************
C*************************************************************************
C Call the matrix inversion routine.
C Invert matrix SIGMA with the inverse stored in RSIG
C*************************************************************************
            CALL MIGS(SIGMA, M, RSIG, INDX)
            PVAR1 = RSIG + MATMUL(TRANSPOSE(AA), AA)
C*************************************************************************
C Call the matrix inversion routine to invert matrix PVAR1 with the
C inverse stored in PVAR
C*************************************************************************
            CALL MIGS(PVAR1, M, PVAR, INDX)
            DO 70 I = 1, N
                  DO 80 J = 1, K
                        BA(J) = Z(I, J) + G(J)
   80             CONTINUE
                  PMEAN1 = MATMUL(TRANSPOSE(AA), BA)
                  PMEAN = MATMUL(PVAR, PMEAN1)
C*************************************************************************
C Call the Cholesky factorization routine. Compute the Cholesky factorization
C of the symmetric definite matrix PVAR and store the C result in RSIG
C*************************************************************************
                  CALL CHFAC (M, PVAR, M, 0.00001, IRANK, RSIG, M)
C*************************************************************************
C Generate a random sample of theta(v) from MVN dist by calling RNMVN
C*************************************************************************
                  CALL RNMVN (1, M, RSIG, M, RTH, 1)
                  DO 90 J = 1, M
                        TH(I, J) = RTH(J) + PMEAN(J)
                        THV(I, J, COUNT) = TH(I, J)
   90             CONTINUE
   70       CONTINUE
C*************************************************************************
C Update samples for item parameters, a(v) and g(v) from their MVN
C posterior distributions
C*************************************************************************
            JJ = 0
            DO 100 J = 1, M
                  DO 110 I = 1, N
                        X(I, 1) = TH(I, J)
                        X(I, 2) = -1
  110             CONTINUE
                  IF (UNIF == 0) THEN
```

```
C*************************************************************************
C Specify the prior means and variances for a(v) and g(v)
C*************************************************************************
                        AMU = 0.0
                        GMU = 0.0
                        AVAR = 1.0
                        GVAR = 1.0
C*************************************************************************
C Put them in vector or matrix format
C*************************************************************************
                        AGMU(1, 1) = AMU
                        AGMU(2, 1) = GMU
                        AGVAR(1, 1) = AVAR
                        AGVAR(2, 2) = GVAR
C*************************************************************************
C Call the matrix inversion routine to invert matrix AGVAR with the
C inverse stored in AGSIG
C*************************************************************************
                        CALL MIGS(AGVAR, 2, AGSIG, INDX)
                        XX = MATMUL(TRANSPOSE(X), X) + AGSIG
                    ELSE IF (UNIF == 1) THEN
                        XX = MATMUL(TRANSPOSE(X), X)
                    END IF
C*************************************************************************
C Call the matrix inversion routine to invert matrix XX with the
C inverse stored in IX
C*************************************************************************
                    CALL MIGS(XX, 2, IX, INDX)
C*************************************************************************
C Call the Cholesky factorization routine. Compute the Cholesky
C factorization of the symmetric definite matrix IX and store the
C result in AMAT
C*************************************************************************
                    CALL CHFAC (2, IX, 2, 0.00001, IRANK, AMAT, 2)
                    JM = 0
                    PRODA = 1.0
  130               JM = JM + 1
                    JJ = JJ + 1
                    DO 120 I = 1, N
                        ZV(I, 1) = Z(I, JJ)
  120               CONTINUE
                    IF (UNIF == 0) THEN
                        XZ = MATMUL(AGSIG, AGMU) + MATMUL(TRANSPOSE(X), ZV)
                    ELSE IF (UNIF == 1) THEN
                       XZ = MATMUL(TRANSPOSE(X), ZV)
                    END IF
                    BZ = MATMUL(IX, XZ)
                    A(JJ) = 0
                    DO WHILE (A(JJ) .LE. 0)
                        CALL RNMVN(1, 2, AMAT, 2, BETA, 1)
                        A(JJ) = BETA(1) + BZ(1, 1)
                        G(JJ) = BETA(2) + BZ(2, 1)
                    END DO
                    AV(COUNT, JJ) = A(JJ)
                    GV(COUNT, JJ) = G(JJ)
                    END IF
```

```
                  PRODA = PRODA*A(JJ)
                  IF (JM .LT. MN(J)) THEN
                        GOTO 130
                  END IF
                  DO 135 I = 1, M
                        C(I, J) = 0.0
  135             CONTINUE
                  C(J, J) = PRODA ** (1/RMN(J))
  100       CONTINUE
C*************************************************************************
C Update samples for the hyperparameter, SIGMA
C*************************************************************************
            CTH = MATMUL(C, TRANSPOSE(TH))
            D = MATMUL(CTH, TRANSPOSE(CTH))
C*************************************************************************
C Call the subroutine to generate the unconstrained covariance matrix
C VAR from the inverse Wishart distribution
C*************************************************************************
            CALL INVWISHRND(D, M, N, VAR)
            DO 140 I = 1, M
            DO 140 J = 1, M
                  SIGMA(I, J) = VAR(I, J)/SQRT(VAR(I, I))/SQRT(VAR(J, J))
                  RHO(I, J, COUNT) = SIGMA(I, J)
  140       CONTINUE
   30 CONTINUE
C*************************************************************************
C Calculate the batch means and se's for a(v), g(v), theta(v) and
C their correlations, and store them in ITEM, PER, and RPER
C*************************************************************************
      BSIZE = (L - BURNIN)/BN
      DO 150 J = 1, K
            COUNT = BURNIN
            TOT1 = 0.0
            TOT2 = 0.0
            SS1 = 0.0
            SS2 = 0.0
            DO 160 JJ = 1, BN
                  SUM1 = 0.0
                  SUM2 = 0.0
                  DO 170 I = 1, BSIZE
                        COUNT = COUNT + 1
                        SUM1 = SUM1 + AV(COUNT, J)
                        SUM2 = SUM2 + GV(COUNT, J)
  170             CONTINUE
                  M1 = SUM1/FLOAT(BSIZE)
                  M2 = SUM2/FLOAT(BSIZE)
                  TOT1 = TOT1 + M1
                  TOT2 = TOT2 + M2
                  SS1 = SS1 + M1*M1
                  SS2 = SS2 + M2*M2
  160       CONTINUE
            ITEM(J, 1) = TOT1/FLOAT(BN)
            ITEM(J, 2) = SQRT((SS1-(TOT1*TOT1/BN))/(BN-1))/SQRT(FLOAT(BN))
            ITEM(J, 3) = TOT2/BN
            ITEM(J, 4) = SQRT((SS2-(TOT2*TOT2/BN))/(BN-1))/SQRT(FLOAT(BN))
```

```
150 CONTINUE
    JJ = 0
    JK = 0
    DO 180 IM = 1, M
          JJ = JK + 1
          JK = JJ + 1
        DO 190 J = 1, N
                COUNT = BURNIN
                TOT3 = 0.0
                SS3 = 0.0
                DO 200 IB = 1, BN
                      SUM3 = 0.0
                      DO 210 I = 1, BSIZE
                            COUNT = COUNT + 1
                            SUM3 = SUM3 + THV(J, IM, COUNT)
210                   CONTINUE
                      M3 = SUM3/FLOAT(BSIZE)
                      TOT3 = TOT3 + M3
                      SS3 = SS3 + M3*M3
200             CONTINUE
          PER(J, JJ) = TOT3/FLOAT(BN)
          PER(J, JK) = SQRT((SS3-(TOT3*TOT3/BN))/(BN-1))/SQRT(FLOAT(BN))
190       CONTINUE
180 CONTINUE
    JK = 0
    DO 220 J = 1, M
    DO 220 IM = J + 1, M
          JK = JK + 1
          COUNT=BURNIN
          TOT = 0.0
          SS = 0.0
          DO 230 JJ = 1, BN
                SUM0 = 0.0
                DO 240 I = 1, BSIZE
                      COUNT = COUNT + 1
                      SUM0 = SUM0 + RHO(J, IM, COUNT)
240             CONTINUE
                M0 = SUM0/FLOAT(BSIZE)
                TOT = TOT + M0
                SS = SS + M0*M0
230       CONTINUE
          RPER(JK, 1) = TOT/FLOAT(BN)
          RPER(JK, 2) = SQRT((SS-(TOT*TOT/BN))/(BN-1))/SQRT(FLOAT(BN))
220 CONTINUE

    RETURN
    END
```

```
      SUBROUTINE INVWISHRND(S, P, V, IW)
C*************************************************************************
C S = p-by-p symmetric, positive definite 'scale' matrix
C P = order of the scale matrix
C V = 'degree of freedom parameter'
C                    (V must be an integer for this routine)
C IW = random matrix from the inverse Wishart distribution
C Note:
C     different sources use different parameterizations w.r.t. V.
C     this routine uses the model that
C     density (IW) is proportional to
C                  exp[-.5*trace(S*inv(IW))]/[det(IW)^((V+p+1)/2)]
C     With this density definition:
C                  mean(IW) = S/(V-p-1)
C*************************************************************************
      INTEGER  P, V, IRANK, INDX(P)
      REAL     S(P, P), IS(P, P), IW(P, P), W(P, P), Z(V, P), ZZ(P, P),
      &        A(P, P), AZ(P, P)

      DO 10 I = 1, V
      DO 10 J = 1, P
      CALL RNNOR (1, Z(I, J))
   10 CONTINUE
      ZZ = MATMUL(TRANSPOSE(Z), Z)
      CALL MIGS(S, P, IS, INDX)
      CALL CHFAC (P, IS, P, 0.00001, IRANK, A, P)
      AZ = MATMUL(TRANSPOSE(A), ZZ)
      W = MATMUL (AZ, A)
      CALL MIGS(W, P, IW, INDX)

      RETURN
      END
```