Summer 7-31-2013

Bayesian Hierarchical Modeling with 3PNO Item Response Models

Yanyan Sheng
Southern Illinois University Carbondale

Todd C. Headrick
Southern Illinois University Carbondale

Published in the American Journal of Mathematics and Statistics, 2013, Vol. 3, No. 5, pp. 281-287. doi:10.5923/j.ajms.20130305.05

American Journal of Mathematics and Statistics 2013, 3(5): 281-287
DOI: 10.5923/j.ajms.20130305.05

Bayesian Hierarchical Modeling with 3PNO Item Response Models

Yanyan Sheng*, **Todd C. Headrick**

Section on Statistics and Measurement, Southern Illinois University, Carbondale, IL, 62901, USA

Abstract Fully Bayesian estimation has been developed for unidimensional IRT models. In this context, prior distributions can be specified in a hierarchical manner so that item hyperparameters are unknown and yet still have their own priors. This type of hierarchical modeling is useful in terms of the three-parameter IRT model as it reduces the difficulty of specifying model hyperparameters that lead to adequate prior distributions. Further, hierarchical modeling ameliorates the noncovergence problem associated with nonhierarchical models when appropriate prior information is not available. As such, a Fortran subroutine is provided to implement a hierarchical modeling procedure associated with the three-parameter normal ogive model for binary item response data using Gibbs sampling. Model parameters can be estimated with the choice of noninformative and conjugate prior distributions for the hyperparameters.

Keywords IRT, Three-parameter Normal Ogive Model, MCMC, Gibbs Sampling, Hyperparameter, Fortran

1. Introduction

The unidimensional item response theory (IRT) model provides a fundamental framework for modeling person-item interaction given the usual assumption of one latent dimension. The popular two-parameter normal ogive (2PNO; e.g.,[1,2]) IRT model specifies that the probability of the i-th person obtaining a correct response on the j-th item as

$$P(y_{ij}) = \Phi(\alpha_j\theta_i - \beta_j) = \int_{-\infty}^{\alpha_j\theta_i-\beta_j} \frac{1}{\sqrt{2\pi}} e^{\frac{-t^2}{2}} dt, \qquad (1)$$

where $i = 1,\ldots,n$ and $j = 1,\ldots,k$. The notations α_j,β_j, and θ_i in (1) are scalar parameters describing the item (i) slope, (ii) intercept, and (iii) person-trait, respectively. Further, the model in (1) assumes that no guessing is involved with respect to the test item responses.

In terms of objective tests that involve multiple-choice or true-or-false items, where an item may be too difficult for some examinees, the three-parameter normal ogive (3PNO; e.g.,[3]) model should be considered. Specifically, the 3PNO model assumes that the probability associated with a correct response is greater than zero even for examinees with very low trait levels, and it is defined as follows

$$P(y_{ij}) = \gamma_j + (1-\gamma_j)\Phi(\alpha_j\theta_i - \beta_j), 0 \le \gamma_j < 1. \qquad (2)$$

Inspection of (2) indicates that the 3PNO model accommodates for guessing by adding the pseudo-chance-level parameter γ_j. As such, the model in (2) is more appealing because it is applicable to a wider variety of testing situations where the 2PNO model may not be appropriate.

In the context of the Bayesian estimation of IRT models, simultaneous estimation of item and person parameters relies on the use of Markov Chain Monte Carlo (MCMC; e.g.,[4,5]) techniques to summarize the posterior distributions. For example, Albert[10] applied a MCMC algorithm (the Gibbs sampler[11]) to the 2PNO model using data augmentation [12], which has been implemented in Fortran[13]. Further, Sahu[14] (see also[15]) generalized this approach to the 3PNO model. However, this generalization, where the model hyperparameters take on specific values (such as in the applications of[16] and[17]), has a disadvantage associated with the nonconvergence problem unless strong informative priors are specified for the item slope and intercept parameters[18].

In the context of the 3PNO model, it has been demonstrated that improper noninformative prior densities for item slope and intercept parameters result in an undefined posterior distribution, which presents the problem of unstable parameter estimates[18, 19]. Further, even with proper informative prior densities, the Gibbs sampling procedure noted above either fails to converge or requires a large number of iterations for the Markov Chain to reach convergence[19]. Sheng[20] indicated that this problem can be resolved by specifying the prior distribution in a hierarchical manner so that the item hyperparameters are unknown and have their own prior distributions. These second-order priors are called hyperpriors and are useful for

* Corresponding author:
ysheng@siu.edu (Yanyan Sheng)
Published online at http://journal.sapub.org/ajms

incorporating uncertainty in the hyperparameters of a prior distribution[6]. Further, it has been demonstrated that a vague hyperprior does not affect posterior precision like a vague prior does[20] because the uncertainty is at the second level of the prior, and consequently its effect on the resulting posterior distribution is so small that the posterior is dominated by the data[21]. This hierarchical modeling approach allows for a more objective approach to inference by estimating the parameters of prior distributions from data rather than specifying them based on subjective information.

Given that MCMC is computationally expensive and that Fortran is fast in terms of numerical computing[22], the purpose of this paper is to provide a subroutine to determine the posterior estimates (and their standard errors) associated with the 3PNO model parameters using Gibbs sampling. The Fortran subroutine has the option of specifying noninformative and conjugate hyperpriors for item slope and intercept parameters.

2. Methodology

2.1. The Gibbs Sampling Procedure

To implement a Gibbs sampling procedure for the 3PNO model defined in (2), a Bernoulli variable W is first introduced such that W_{ij} = 1 (or W_{ij} = 0) if the i-th person knows (or does not know) the correct answer to the j-th item. The probability function associated with W_{ij} is defined as[14]

$$P(W_{ij} = w_{ij}) = \Phi(\eta_{ij})^{w_{ij}}(1-\Phi(\eta_{ij}))^{1-w_{ij}} \tag{3}$$

where $\eta_{ij} = \alpha_j\theta_i - \beta_j$. As such, if W_{ij} = 0 then the i-th person will guess the j-th item correctly (or incorrectly) with a probability γ_j (or $(1-\gamma_j)$). Further, a latent random variable Z is introduced such that $Z_{ij} \sim N(\eta_{ij}, 1)$[10, 12] where if $W_{ij} = 1$ (or $W_{ij} = 0$) then $Z_{ij} > 0$ (or $Z_{ij} \le 0$). The prior distributions associated with the item and person parameters are assumed to be as follows

$$\theta_i \sim N(\mu, \sigma^2),\ \alpha_j \sim N_{(0,\infty)}(\mu_\alpha, \sigma_\alpha^2),\ \beta_j \sim N(\mu_\beta, \sigma_\beta^2),$$
$$\gamma_j \sim Beta(s,t).$$

Note that we consider models where the prior distributions are assumed for the hyperparameters μ_α, μ_β, σ^2_α and σ^2_β, instead of specifying values for them.

The joint posterior distribution of ($\boldsymbol{\theta}$, $\boldsymbol{\xi}$, $\boldsymbol{\gamma}$, $\mathbf{W}$, $\mathbf{Z}$, $\boldsymbol{\mu}_\xi$, $\boldsymbol{\Sigma}_\xi$), where $\boldsymbol{\xi}_j = (\alpha_j, \beta_j)'$, $\boldsymbol{\mu}_\xi = (\mu_\alpha, \mu_\beta)'$, $\boldsymbol{\Sigma}_\xi = \text{diag}(\sigma^2_\alpha, \sigma^2_\beta)$, is

$$\begin{aligned} p(\boldsymbol{\theta},\boldsymbol{\xi},\boldsymbol{\gamma},\mathbf{W},\mathbf{Z},\boldsymbol{\mu}_\xi,\boldsymbol{\Sigma}_\xi \mid \mathbf{y}) \propto f(\mathbf{y}\mid\mathbf{W},\boldsymbol{\gamma})p(\mathbf{W}\mid\mathbf{Z}) \\ p(\mathbf{Z}\mid\boldsymbol{\theta},\boldsymbol{\xi})p(\boldsymbol{\theta})p(\boldsymbol{\xi}\mid\boldsymbol{\mu}_\xi,\boldsymbol{\Sigma}_\xi)p(\boldsymbol{\gamma})p(\boldsymbol{\mu}_\xi)p(\boldsymbol{\Sigma}_\xi) \end{aligned} \tag{4}$$

where $f(\mathbf{y}|\mathbf{W}, \boldsymbol{\gamma})$ is the likelihood function.

The full conditional distribution of W_{ij}, Z_{ij}, θ_i, ξ_j, and γ_j can be derived in closed form as follows:

$$W_{ij} \mid \bullet \sim \begin{cases} Bernoulli\left(\dfrac{\Phi(\eta_{ij})}{\gamma_j + (1-\gamma_j)\Phi(\eta_{ij})}\right), & if\ y_{ij} = 1 \\ Bernoulli(0), & if\ y_{ij} = 0 \end{cases}; \tag{5}$$

$$Z_{ij} \mid \bullet \sim \begin{cases} N_{(0,\infty)}(\eta_{ij},1), & if\ W_{ij} = 1 \\ N_{(-\infty,0)}(\eta_{ij},1), & f\ W_{ij} = 0 \end{cases}; \tag{6}$$

$$\theta_i \mid \bullet \sim N\left(\frac{\sum_j (Z_{ij}+\beta_j)\alpha_j + \mu/\sigma^2}{1/\sigma^2 + \sum_j \alpha_j^2}, \frac{1}{1/\sigma^2 + \sum_j \alpha_j^2}\right), \tag{7}$$

$$\boldsymbol{\xi}_j \mid \bullet \sim N\left((\mathbf{x}'\mathbf{x}+\boldsymbol{\Sigma}_\xi^{-1})^{-1}(\mathbf{x}'\mathbf{Z}_j + \boldsymbol{\Sigma}_\xi^{-1}\boldsymbol{\mu}_\xi), (\mathbf{x}'\mathbf{x}+\boldsymbol{\Sigma}_\xi^{-1})^{-1}\right)I(\alpha_j > 0) \tag{8}$$

where $\mathbf{x} = [\boldsymbol{\theta}, -1]$;

$$\gamma_j \mid \bullet \sim Beta(a_j + s, b_j - a_j + t), \tag{9}$$

where a_j is the number of correct responses obtained by guessing, and b_j is the number of persons who do not know the correct answer to the j-th item.

In terms of the hyperparameters μ_α, μ_β, σ^2_α and σ^2_β, their full conditional distributions are

$$\mu_\alpha \mid \bullet \sim N(\textstyle\sum_j \alpha_j/k, \sigma_\alpha^2/k), \tag{10}$$

$$\mu_\beta \mid \bullet \sim N(\textstyle\sum_j \beta_j/k, \sigma_\beta^2/k), \tag{11}$$

$$\sigma_\alpha^2 \mid \bullet \sim inv-gamma(k/2, \textstyle\sum_j(\alpha_j-\mu_\alpha)^2/2), \tag{12}$$

$$\sigma_\beta^2 \mid \bullet \sim inv-gamma(k/2, \textstyle\sum_j(\beta_j-\mu_\beta)^2/2), \tag{13}$$

respectively, with uniform noninformative hyperpriors $p(\mu_\alpha, \sigma_\alpha^2) \propto 1/\sigma_\alpha^2$ and $p(\mu_\beta, \sigma_\beta^2) \propto 1/\sigma_\beta^2$, or as

$$\mu_\alpha \mid \bullet \sim N\left(\frac{\sum_j \alpha_j/\sigma_\alpha^2}{1/\tau_\alpha + k/\sigma_\alpha^2}, \frac{1}{1/\tau_\alpha + k/\sigma_\alpha^2}\right), \tag{14}$$

$$\mu_\beta \mid \bullet \sim N\left(\frac{\sum_j \beta_j/\sigma_\beta^2}{1/\tau_\beta + k/\sigma_\beta^2}, \frac{1}{1/\tau_\beta + k/\sigma_\beta^2}\right), \tag{15}$$

$$\sigma_\alpha^2 \mid \bullet \sim inv-gamma(\varepsilon_1 + k/2, \varepsilon_2 + \textstyle\sum_j(\alpha_j-\mu_\alpha)^2/2), \tag{16}$$

$$\sigma_\beta^2 \mid \bullet \sim inv-gamma(\varsigma_1 + k/2, \varsigma_2 + \textstyle\sum_j(\beta_j-\mu_\beta)^2/2), \tag{17}$$

with conjugate hyperprior distributions $\mu_\alpha \sim N(0, \tau_\alpha)$, $\mu_\beta \sim N(0, \tau_\beta)$, $\sigma_\alpha^2 \sim inv-gamma(\varepsilon_1, \varepsilon_2)$,

$\sigma_\beta^2 \sim inv-gamma(\varsigma_1, \varsigma_2)$.

As such, using starting values of $\theta^{(0)}$, $\boldsymbol{\xi}^{(0)}$, $\gamma^{(0)}$, $\boldsymbol{\mu}_\xi^{(0)}$ and $\boldsymbol{\Sigma}_\xi^{(0)}$, observations ($W^{(l)}$, $Z^{(l)}$, $\theta^{(l)}$, $\boldsymbol{\xi}^{(l)}$, $\gamma^{(l)}$, $\boldsymbol{\mu}_\xi^{(l)}$, $\boldsymbol{\Sigma}_\xi^{(l)}$) can be simulated from the Gibbs sampler by iteratively drawing from their respective full conditional distributions specified in (5) through (15). The transition from ($W^{(l-1)}$, $Z^{(l-1)}$, $\theta^{(l-1)}$, $\boldsymbol{\xi}^{(l-1)}$, $\gamma^{(l-1)}$, $\boldsymbol{\mu}_\xi^{(l-1)}$, $\boldsymbol{\Sigma}_\xi^{(l-1)}$) to ($W^{(l)}$, $Z^{(l)}$, $\theta^{(l)}$, $\boldsymbol{\xi}^{(l)}$, $\gamma^{(l)}$, $\boldsymbol{\mu}_\xi^{(l)}$, $\boldsymbol{\Sigma}_\xi^{(l)}$) is based on the following seven steps:

1. Draw $W^{(l)} \sim p(W|\mathrm{y}, \theta^{(l-1)}, \boldsymbol{\xi}^{(l-1)}, \gamma^{(l-1)})$;
2. Draw $Z^{(l)} \sim p(Z|W^{(l)}, \theta^{(l-1)}, \boldsymbol{\xi}^{(l-1)})$;
3. Draw $\theta^{(l)} \sim p(\theta|Z^{(l)}, \boldsymbol{\xi}^{(l-1)})$;
4. Draw $\boldsymbol{\xi}^{(l)} \sim p(\boldsymbol{\xi}|Z^{(l)}, \theta^{(l)}, \boldsymbol{\mu}_\xi^{(l-1)}, \boldsymbol{\Sigma}_\xi^{(l-1)})$;
5. Draw $\gamma^{(l)} \sim p(\gamma|\mathrm{y}, W^{(l)})$;
6. Draw $\boldsymbol{\mu}_\xi^{(l)} \sim p(\boldsymbol{\mu}_\xi|\boldsymbol{\xi}^{(l)}, \boldsymbol{\Sigma}_\xi^{(l-1)})$;
7. Draw $\boldsymbol{\Sigma}_\xi^{(l)} \sim p(\boldsymbol{\Sigma}_\xi|\boldsymbol{\xi}^{(l)}, \boldsymbol{\mu}_\xi^{(l)})$.

This iterative procedure produces a sequence of ($\theta^{(l)}$, $\boldsymbol{\xi}^{(l)}$, $\gamma^{(l)}$), l = 1,…, 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 the item parameters $\boldsymbol{\xi}$, γ and person parameters θ. As with standard Monte Carlo, with large enough samples, the posterior means of $\boldsymbol{\xi}$, γ and θ are considered as estimates of the true parameters. However, their Monte Carlo standard errors cannot be calculated using the sample standard deviations because subsequent samples in each Markov chain are autocorrelated (e.g.[10, 23]). One approach to calculating the standard errors is through batching[24]. Specifically, with a long chain of samples being separated into contiguous batches of equal length, the Monte Carlo standard deviation for each parameter is then estimated to be the standard deviation of these batched means. And the Monte Carlo standard error of the estimate is a ratio of the Monte Carlo standard deviation and the square root of the number of batches.

2.2. The Fortran Subroutine

The subroutine (see the Appendix) initially sets the starting values for the parameters such that $\theta_i^{(0)} = 0$, $\alpha_j^{(0)} = 1$, $\beta_j^{(0)} = 1$, $\gamma_j^{(0)} = .2$,[16], and, $\mu_\alpha^{(0)} = \mu_\beta^{(0)} = 0$, $\sigma^2_\alpha{}^{(0)} = \sigma^2_\beta{}^{(0)} =1$. The subroutine then iteratively draws random samples for W, Z, θ, $\boldsymbol{\xi}$, and γ from their respective full conditional distributions specified in (5) through (9) with $\mu = 0$, $\sigma^2 =1$, and $s = 5$, $t = 17$. Samples associated with the hyperparameters for $\boldsymbol{\xi}$ are simulated from either (10) through (13), where uniform noninformative priors are assumed for $\boldsymbol{\mu}_\xi$ and $\boldsymbol{\Sigma}_\xi$, or from (14) through (17), where conjugate priors are adopted for them with $\tau_\alpha = \tau_\beta = 100$, $\varepsilon_1 = \varsigma_1 = 2$, and $\varepsilon_2 = \varsigma_2 = .001$. We would note that the conjugate priors specified in this manner are weakly informative. The algorithm continues until all the L samples are simulated. It then discards the early burn-in samples, and computes the posterior estimates and standard errors for the model parameters, θ, α, β, and γ, using the batching scheme described above.

Table 1. Posterior estimates and Monte Carlos standard errors (MCSEs) for α with noninformative and conjugate priors assumed for $\boldsymbol{\mu}_\xi$ and $\boldsymbol{\Sigma}_\xi$

	Posterior estimates			
	Noninformative priors		Conjugate priors	
Parameter	Estimate	MCSE	Estimate	MCSE
.0966	.0835	.0018	.0901	.0028
.0971	.0860	.0039	.0818	.0023
.4589	.4671	.0048	.4669	.0036
.9532	.9197	.0136	.9091	.0131
.0771	.1062	.0044	.1128	.0033
.4891	.4459	.0060	.4885	.0157
.8599	.8033	.0229	.7553	.0085
.9427	.8570	.0238	.9288	.0416
.2727	.3496	.0079	.3577	.0107
.6532	.6669	.0200	.6805	.0131

Table 2. Posterior estimates and Monte Carlos standard errors (MCSEs) for β with noninformative and conjugate priors assumed for $\boldsymbol{\mu}_\xi$ and $\boldsymbol{\Sigma}_\xi$

	Posterior estimates			
	Noninformative priors		Conjugate priors	
Parameter	Estimate	MCSE	Estimate	MCSE
-.7997	-.8640	.0182	-.7657	.0376
-.5321	-.5792	.0237	-.5532	.0218
.8583	.8638	.0133	.8623	.0057
.7237	.7111	.0114	.6913	.0091
-.8184	-.8571	.0238	-.7776	.0386
-.5834	-.8935	.0151	-.7923	.0339
.3629	.1315	.0231	.0922	.0124
-.9010	-.9853	.0238	-.9255	.0320
-.9339	-.8394	.0225	-.7953	.0341
-.3978	-.4629	.0292	-.4424	.0210

Table 3. Posterior estimates and Monte Carlos standard errors (MCSEs) for γ with noninformative and conjugate priors assumed for $\boldsymbol{\mu}_\xi$ and $\boldsymbol{\Sigma}_\xi$

	Posterior estimates			
	Noninformative priors		Conjugate priors	
Parameter	Estimate	MCSE	Estimate	MCSE
.3497	.2639	.0144	.3445	.0268
.2913	.2636	.0169	.2891	.0164
.0473	.0550	.0035	.0550	.0014
.0497	.0511	.0021	.0458	.0017
.3113	.2582	.0213	.3226	.0316
.4948	.2735	.0138	.3413	.0193
.2453	.1486	.0105	.1316	.0061
.4687	.3708	.0243	.4189	.0262
.1720	.2739	.0198	.3114	.0265
.3001	.2472	.0205	.2615	.0143

For example, for a 4000-by-10 (i.e., $n = 4{,}000$ and $k = 10$) dichotomous (0-1) data matrix simulated using the item parameters shown in the first column of Tables 1 to 3, the Gibbs sampler was implemented 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.

Two sets of the posterior means for α, β, and γ, as well as their Monte Carlo standard errors, were obtained assuming the noninformative or weakly informative hyperpriors described previously, and are displayed in the rest of the tables. We note that the item parameters were estimated with enough accuracy and the two sets of posterior estimates differ only slightly from each other, signifying that the results are not sensitive to the choice of prior distributions for the hyperparameters $\boldsymbol{\mu}_\xi$ and $\boldsymbol{\Sigma}_\xi$. In addition, the small values of the Monte Carlo standard errors suggested that the Markov chains with a run length of 10,000 and a burn-in period of 5,000 reached the stationary distribution.

3. Conclusions

The Fortran subroutine leaves it to the user to choose between uniform and conjugate priors for the hyperparameters for item slope and intercept parameters, $\boldsymbol{\mu}_\xi$ or $\boldsymbol{\Sigma}_\xi$. Further, the user can change the source code so that the prior distribution for θ_i assumes a different location μ, or scale σ^2. Similarly, the values of s and t can be modified to reflect different prior beliefs on the distribution for the pseudo-chance parameter. One can also change the values for τ_α, τ_β, ε_1, ε_2, ς_1 or ς_2 to specify different prior densities for the hyperparameters $\boldsymbol{\mu}_\xi$ and $\boldsymbol{\Sigma}_\xi$. It is noted that convergence can be assessed by comparing the marginal posterior mean and standard deviation of each parameter computed for every 1,000 samples after the burn-ins. Similar values provide a rough indication of similar marginal posterior densities, which further indicates possible convergence of the Gibbs sampler[25, 26].

Appendix

```
SUBROUTINE GSU3(Y, N, K, L, BURNIN, BN, PRIOR, ITEM,
PERSON)
c*******************************************c
c Y is the n-by-k binary item response data   c
c N is the number of subjects                 c
c K is the test length (number of items)      c
c L is the number of iterations using Gibbs   c
c    sampling                                 c
c BURNIN is the first number of iterations    c
c   that are to be discarded                  c
c BN is the number of batches                 c
c PRIOR is a 1-2 indicator with 1 = uniform   c
c   priors for item slope & intercept hyper-  c
c   parameters and 2 = conjugate priors       c
c ITEM is a k-by-6 matrix of posterior        c
c   estimates and standard errors for item    c
c   parameters                                c
c PERSON is a n-by-2 matrix of posterior      c
c   estimates and standard errors for person  c
c   abilities                                 c
c*******************************************c
    INTEGER  L, COUNT, IRANK, Y(N,K), BURNIN,
    &   PRIOR, INDX(2), BN, BSIZE, W(N,K)
    REAL A(K), G(K), TH(N), LP, MU, VAR,
    &   AV(L,K), GV(L,K),THV(N,L), S, T, U,
    &   Z(N,K), V, MN, MSUM, PVAR,PMEAN, TT,
    &   X(N,2), XX(2,2), IX(2,2), ZV(N,1),
    &   XZ(2,1), AMAT(2,2), BZ(2,1), AMU,
    &   GMU, AVAR, GVAR, AGMU(2,1), AGVAR(2,2),
    &   SIGMA(2,2), BETA(1,2), BI(1,2),
    &   ITEM(K,4),PERSON(N,2), SUM1, SUM2,
    &   SUM3, SUM4, M1, M2, M3, M4, TOT1, TOT2,
    &   TOT3, TOT4, SS1, SS2, SS3, SS4, PIN,
    &   QIN, GP2, SA, SG, SSA, SSG, AMM, GMM,
    &   AMV, GMV, AV1, AV2, GV1, GV2, RAG(2)
   REAL C(K), NP, P, CV(L,K), IR, SD, TD, AP0,
    &   GP0, AP1, GP1, AP2
```

```
    DOUBLE PRECISION   BB, TMP
c*********************************************c
c Connect to external libraries for normal    c
c (RNNOR), uniform (RUN), beta (RNBET) and   c
c gamma (RNGAM) random number generator,      c
c inverse (ANORIN, DNORIN) and CDF (ANORDF,   c
c DNORDF) for the standard normal distribution c
c and Cholesky factorization (CHFAC) routines c
c*********************************************c
    EXTERNAL  RNNOR, RNSET, RNUN, ANORDF, ANORIN,
    &    CHFAC, DNORDF, DNORIN, RNBET, SSCAL,
    &    RNGAM
c*********************************************c
c Set initial values for item parameters      c
c alpha(A), beta(G), gamma(C), and person     c
c abilities theta(TH) so that alpha=1, beta=0, c
c gamma=.2 for all k items, and theta=0 for   c
c all n persons                               c
c*********************************************c
    DO 10 I = 1, K
        A(I) = 1.0
        G(I) = 0.0
        C(I) = 0.2
  10 CONTINUE
    DO 20 I=1, N
        TH(I)=0.0
  20 CONTINUE
c*********************************************c
c Set initial values for the prior means and  c
c variances for alpha and beta                c
c*********************************************c
    AMU = 0.0
    GMU = 0.0
    AVAR = 1.0
    GVAR = 1.0
c*********************************************c
c MU and VAR are the mean and standard        c
c deviation for the prior distribution of     c
c theta                                       c
c*********************************************c
    MU = 0.0
    VAR = 1.0
c*********************************************c
c Specify the hyperparameters in the conjugate c
c prior distributions for AMU, GMU, AVAR and  c
c GVAR                                        c
c*********************************************c
    IF (PRIOR == 2) THEN
        AP0 = 100.0
        GP0 = 100.0
        AP1 = 2.0
        GP1 = 2.0
        AP2 = 0.001
        GP2 = 0.001
    END IF
c*********************************************c
c Set values for the hyperparameters (SD, TD) c
```

```
c in the beta prior distribution for gamma      c
c*********************************************c
    SD = 5.0
    TD = 17.0
c*********************************************c
c Start iteration                               c
c*********************************************c
    COUNT = 0
    DO 30 IT = 1, L
        COUNT = COUNT + 1
c*********************************************c
c Update samples for W and Z from their         c
c posterior distributions                       c
c*********************************************c
        DO 40 I = 1, N
        DO 40 J = 1, K
            W(I,J)=0.0
            LP = TH(I)*A(J) - G(J)
            NP = ANORDF(LP)
            P = NP/(NP + C(J)*(1-NP))
            IF (Y(I,J) == 1) THEN
                CALL RNBIN(1,1,P,IR)
                W(I,J) = IR
            END IF
            BB = ANORDF((0.0-LP))
            CALL RNUN (1, U)
            TMP = (BB*(1-W(I,J))+
                  (1-BB)*W(I,J))*U+BB*W(I,J)
            Z(I,J) = DNORIN(TMP) + LP
  40    CONTINUE

c*********************************************c
c Update samples for theta from its normal      c
c posterior distributions                       c
c*********************************************c
        V = 1/SUM(A*A)
        PVAR = 1/(1/V + 1/VAR)
        DO 50 I = 1, N
            MSUM = 0.0
            DO 60 J = 1, K
              MSUM = MSUM + A(J)*(Z(I,J) + G(J))
  60        CONTINUE
            MN = MSUM*V
            PMEAN = (MN/V + MU/VAR)*PVAR
            CALL RNNOR(1,TT)
            TH(I) = TT*SQRT(PVAR) + PMEAN
            THV(I,COUNT) = TH(I)
  50   CONTINUE
c*********************************************c
c Update samples for item parameters, alpha     c
c and beta, from their multivariate normal      c
c posterior distributions                       c
c*********************************************c
         DO 70 J = 1, 1
         DO 70 I = 1, N
            X(I,J) = TH(I)
  70     CONTINUE
         DO 80 J = 2, 2
         DO 80 I = 1, N
            X(I,J) = -1
  80     CONTINUE
c*********************************************c
c Put the prior item means and variances in     c
c vector or matrix format                       c
c*********************************************c
         AGMU(1,1) = AMU
         AGMU(2,1) = GMU
         AGVAR(1,1) = AVAR
         AGVAR(2,2) = GVAR
c*********************************************c
c Call the matrix inversion routine.            c
c Invert matrix AGVAR with the inverse          c
c stored in SIGMA                               c
c*********************************************c
         CALL MIGS(AGVAR, 2, SIGMA, INDX)
         XX = MATMUL(TRANSPOSE(X),X)+SIGMA
c*********************************************c
c Call the matrix inversion routine.            c
c Invert matrix XX with the inverse stored in c
c IX                                            c
c*********************************************c
         CALL MIGS(XX,2,IX,INDX)
c*********************************************c
c Call the Cholesky factorization routine.      c
c Compute the Cholesky factorization of the     c
c symmetric definite matrix IX and store the  c
c result in AMAT                                c
c*********************************************c
CALL CHFAC (2, IX, 2, 0.00001, IRANK, AMAT, 2)
         DO 90 J = 1, K
            DO 100 I = 1, N
                ZV(I,1) = Z(I,J)
 100        CONTINUE
            XZ = MATMUL(SIGMA,AGMU)+
            MATMUL(TRANSPOSE(X),ZV)
            BZ = MATMUL(IX,XZ)
            A(J) = 0
            DO WHILE (A(J).LE.0)
                CALL RNNOR (2, BI)
        BETA = MATMUL(BI,AMAT) + TRANSPOSE(BZ);
                A(J) = BETA(1,1)
                G(J) = BETA(1,2)
            END DO
            AV(COUNT,J) = A(J)
            GV(COUNT,J) = G(J)
  90    CONTINUE
c*********************************************c
c Update samples for gamma from its beta        c
c posterior distributions                       c
c*********************************************c
        DO 110 J = 1, K
            T = 0.0
            S = 0.0
            DO 120 I = 1, N
```

```
                IF (W(I,J) == 0) THEN
                    T = T + 1
                    S = S + Y(I,J)
                END IF
 120        CONTINUE
            PIN = S + SD
            QIN = T - S + TD
            CALL RNBET (1, PIN, QIN, TT)
            C(J) = TT
            CV(COUNT,J) = TT
 110    CONTINUE
c*********************************************c
c Update samples for the hyperparameters AMU, c
c GMU, AVAR and GVAR from  their posterior    c
c distributions                               c
c*********************************************c
        SA = 0.0
        SG = 0.0
        SSA = 0.0
        SSG = 0.0
        DO 130 J = 1, K
            SA = SA + A(J)
            SG = SG + G(J)
            SSA = SSA + (A(J)-AMU)*(A(J)-AMU)
            SSG = SSG + (G(J)-GMU)*(G(J)-GMU)
 130    CONTINUE
        IF (PRIOR == 1) THEN
            AMV = 1.0/(K/AVAR)
            GMV = 1.0/(K/GVAR)
            AMM = AMV*SA/AVAR
            GMM = GMV*SG/GVAR
            AV1 = K/2.0
            GV1 = K/2.0
            AV2 = SSA/2.0
            GV2 = SSG/2.0
        ELSE IF (PRIOR == 2) THEN
            AMV = 1.0/(K/AVAR+1.0/AP0)
            GMV = 1.0/(K/GVAR+1.0/GP0)
            AMM = AMV*SA/AVAR
            GMM = GMV*SG/GVAR
            AV1 = K/2.0 + AP1
            GV1 = K/2.0 + GP1
            AV2 = SSA/2.0 + AP2
            GV2 = SSG/2.0 + GP2
        END IF
        CALL RNNOR(2,RAG)
        AMU = RAG(1)*SQRT(AMV) + AMM
        GMU = RAG(2)*SQRT(GMV) + GMM
        CALL RNGAM (1, AV1, TT)
        CALL SSCAL (1, AV2, TT, 1)
        AVAR = 1.0/TT
        CALL RNGAM (1, GV1, TT)
        CALL SSCAL (1, GV2, TT, 1)
        GVAR = 1.0/TT
  30 CONTINUE
c*********************************************c
```

```
c Calculate the batch means and mcse's for    c
c alpha, beta, gamma and theta and store them c
c in ITEM and PERSON                          c
c*********************************************c
    BSIZE = (L-BURNIN)/BN

    DO 200 J = 1, K
        COUNT = BURNIN
        TOT1 = 0.0
        TOT2 = 0.0
        TOT3 = 0.0
        SS1 = 0.0
        SS2 = 0.0
        SS3 = 0.0
        DO 210 M = 1, BN
            SUM1 = 0.0
            SUM2 = 0.0
            SUM3 = 0.0
            DO 220 I = 1, BSIZE
                COUNT = COUNT + 1
                SUM1 = SUM1 + AV(COUNT,J)
                SUM2 = SUM2 + GV(COUNT,J)
                SUM3 = SUM3 + CV(COUNT,J)
 220            CONTINUE
                M1 = SUM1/BSIZE
                M2 = SUM2/BSIZE
                M3 = SUM3/BSIZE
                TOT1 = TOT1 + M1
                TOT2 = TOT2 + M2
                TOT3 = TOT3 + M3
                SS1 = SS1 + M1*M1
                SS2 = SS2 + M2*M2
                SS3 = SS3 + M3*M3
 210        CONTINUE
        ITEM(J,1) = TOT1/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))
        ITEM(J,5) = TOT3/BN
        ITEM(J,6) = SQRT((SS3-(TOT3*TOT3/BN))/
                      (BN-1))/SQRT(FLOAT(BN))
 200    CONTINUE

    DO 230 J = 1, N
        COUNT = BURNIN
        TOT4 = 0.0
        SS4 = 0.0
        DO 240 M = 1, BN
            SUM4 = 0.0
            DO 250 I = 1, BSIZE
                COUNT = COUNT + 1
                SUM4 = SUM4 + THV(J,COUNT)
 250            CONTINUE
            M4 = SUM4/BSIZE
            TOT4 = TOT4 + M4
```

```
          SS4 = SS4 + M4*M4
 240     CONTINUE
        PERSON(J,1) = TOT4/BN
        PERSON(J,2)=SQRT((SS4-(TOT4*TOT4/BN))/
                    (BN-1))/SQRT(FLOAT(BN))
 230    CONTINUE

    RETURN
    END
```

REFERENCES

[1] Lord, F. M., 1952, A theory of test scores, Psychometric Monograph No. 7.

[2] Lord, F. M., and Novick, M. R., 1968, Statistical theories of mental test scores, Addison-Wesley, Reading, MA.

[3] Lord, F. M., 1980, Applications of item response theory to practical testing problems, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[4] Smith, A. F. M., and Roberts, G. O., 1993, Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods, Journal of the Royal Statistical Society, Series B, 55, 3–23.

[5] Tierney, L., 1994, Markov chains for exploring posterior distributions, The Annals of Statistics, 22, 1701–1762.

[6] Carlin, B. P., and Louis, T. A., 2000, Bayes and empirical Bayes methods for data analysis (2nd ed.), Chapman & Hall, London.

[7] Chib, S., and Greenberg, E., 1995, Understanding the Metropolis-Hastings algorithm, The American Statistician, 49(4), 327–335.

[8] Gelfand, A. E., and Smith, A. F. M., 1990, Sampling-based approaches to calculating marginal densities, Journal of the American Statistical Association, 85, 398–409.

[9] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B., 2003, Bayesian data analysis, Chapman & Hall/CRC, Boca Raton.

[10] Albert, J. H., 1992, Bayesian estimation of normal ogive item response curves using Gibbs sampling, Journal of Educational Statistics, 17, 251–269.

[11] Geman, S., and Geman, D., 1984, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Trans. Pattern Analysis and Machine Intelligence, 6, 721–741.

[12] Tanner, M. A., and Wong, W. H., 1987, The calculation of posterior distribution by data augmentation (with discussion), Journal of the American Statistical Association, 82, 528–550.

[13] Sheng, Y., and Headrick, T. C., 2007, An algorithm for implementing Gibbs sampling for 2PNO IRT models, Journal of Modern Applied Statistical Methods, 6, 341–349.

[14] Sahu, S. K., 2002, Bayesian estimation and model choice in item response models, Journal of Statistical Computation and Simulation, 72, 217–232.

[15] Johnson, V. E., and Albert, J. H., 1999, Ordinal data modeling, Springer-Verlag, New York.

[16] Béguin, A. A., and Glas, C. A. W., 2001, MCMC estimation and some model-fit analysis of multidimensional IRT models, Psychometrika, 66, 541–562.

[17] Glas, C. W., and Meijer, R. R., 2003, A Bayesian approach to person fit analysis in item response theory models, Applied Psychological Measurement, 27, 217–233.

[18] Sheng, Y., 2008, Markov chain Monte Carlo estimation of normal ogive IRT models in MATLAB, Journal of Statistical Software, 25(8), 1–15.

[19] Sheng, Y., 2010, A sensitivity analysis of Gibbs sampling for 3PNO IRT models: Effects of prior specifications on parameter estimates, Behaviormetrika, 37(2), 87–110.

[20] Sheng, Y., 2013, An empirical investigation of Bayesian hierarchical modeling with unidimensional IRT models, Behaviormetrika, 40(1), 19–40.

[21] Williams, C. L., and Locke, A., 2003, Hyperprior imprecision in hierarchical Bayesian modeling of cluster Bernoulli observations, InterStat: Statistics on the Internet. URL: http://interstat.statjournals.net/YEAR/2003/abstracts/0310001.php.

[22] Brainerd, W., 2003, The importance of Fortran in the 21st century, Journal of Modern Statistical Methods, 2, 14–15.

[23] Patz, R. J., and 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.

[24] Ripley, B. D., 1987, Stochastic simulation, Wiley, New York.

[25] Gelfand, A. E., Hills, S. E., Racine-Poon, A., and 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.

[26] Hoijtink, H., and Molenaar, I. W., 1997, A multidimensional item response model: Constrained latent class analysis using posterior predictive checks, Psychometrika, 62, 171–189.