Bayesian Hierarchical Modeling with 3PNO Item Response Models

yanyan Sheng, Southern Illinois University Carbondale
Todd Christopher Headrick, Southern Illinois University Carbondale

Available at: https://works.bepress.com/todd_headrick/26/
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 nonconvergence 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

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

where $i = 1,\ldots,n$ and $j = 1,\ldots,k$. The notations $\alpha_j,\beta_j, \text{and} \theta_i$ 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

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

Inspection of (2) indicates that the 3PNO model accommodates for guessing by adding the pseudo-chance-level parameter $\gamma_j$. 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...
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 \( W_j \) is introduced such that \( Z_{ij} \) is introduced such that

\[
Z_{ij} \sim \begin{cases} 
N(\alpha_j, \sigma^2_a) & \text{if } W_j = 1 \\
N(\mu, \sigma^2) & \text{if } W_j = 0
\end{cases} 
\]

where \( \alpha_j = \alpha_i \theta_j - \beta_j \). As such, if \( W_j = 0 \) then the \( i \)-th person will guess the \( j \)-th item correctly (or incorrectly) with a probability \( \gamma_j \) (or \( 1 - \gamma_j \)). Further, a latent random variable \( Z_{ij} \) is introduced such that \( Z_{ij} \sim \mathcal{N}(\alpha_j, \sigma^2_a) \)

\[
\\gamma_j \sim \text{Beta}(s, t)
\]

Note that we consider models where the prior distributions are assumed for the hyperparameters \( \mu_a, \mu_B, \sigma^2_a \) and \( \sigma^2_B \), instead of specifying values for them.

The joint posterior distribution of \( (\theta_i, \xi_j, \gamma, W, Z, \mu_a, \Sigma) \), where \( \xi \) = \( (\alpha_i, \beta_j, \mu, \sigma^2_a, \sigma^2_B) \), is

\[
p(\theta_i, \xi_j, \gamma, W, Z, \mu_a, \Sigma) \propto f(y | W, \gamma)p(W)p(Z | 0, \Sigma)p(\theta | \mu, \sigma^2_a, \sigma^2_B)
\]

where \( f(y | W, \gamma) \) is the likelihood function.

The full conditional distribution of \( W_{ij}, Z_{ij}, \theta_i, \xi_j, \gamma_j \), and \( \gamma_j \) can be derived in closed form as follows:

\[
W_{ij} \sim \begin{cases} 
\text{Bernoulli} \left( \frac{\Phi(\eta_{ij})}{\Phi(\eta_{ij}) + \Phi(-\eta_{ij})} \right) & \text{if } y_{ij} = 1 \\
\text{Bernoulli}(0) & \text{if } y_{ij} = 0
\end{cases}
\]

\[
\xi_j \sim \mathcal{N}(x'x + \Sigma^{-1})^{-1}(x'z_j + \Sigma^{-1} \mu_a), (x'x + \Sigma^{-1})^{-1} \eta(\gamma_j > 0)
\]

where \( x = \{0, 1\} \); 

\[
\gamma_j \sim \text{Beta}(a + s, b_j - a_j + t),
\]

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 \( \mu_a, \mu_B, \sigma^2_a \) and \( \sigma^2_B \), their full conditional distributions are

\[
\mu_a \sim \mathcal{N} \left( \sum_{i=1}^{l} \alpha_i / k, \sigma^2_a / k \right)
\]

\[

\mu_B \sim \mathcal{N} \left( \sum_{j=1}^{l} \beta_j / k, \sigma^2_B / k \right)
\]

\[
\sigma^2_a \sim \text{inv-gamma}(k/2, \sum_{i=1}^{l} (\alpha_i - \mu_a)^2 / 2)
\]

\[
\sigma^2_B \sim \text{inv-gamma}(k/2, \sum_{j=1}^{l} (\beta_j - \mu_B)^2 / 2)
\]

respectively, with uniform noninformative hyperpriors

\[
p(\mu_a, \sigma^2_a) \propto 1 / \sigma^2_a
\]

and

\[
p(\mu_B, \sigma^2_B) \propto 1 / \sigma^2_B
\]

as

\[
\mu_a \sim \mathcal{N}(0, \tau_a); \quad \mu_B \sim \mathcal{N}(0, \tau_B); \quad \sigma^2_a \sim \text{inv-gamma}(\varsigma_1, \varsigma_2)
\]

\[
\sigma^2_B \sim \text{inv-gamma}(\varsigma_3, \varsigma_4)
\]

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

1. Draw \( W_{ij} \sim p(W_{ij} | \theta^{(i)}, \xi^{(i)}, \gamma^{(i)}) \);
2. Draw \( Z_{ij} \sim p(Z_{ij} | W_{ij}, \theta^{(i)}, \xi^{(i)}) \);
3. Draw \( \theta^{(i)} \sim p(\theta^{(i)} | Z_{ij}, \xi^{(i)}) \);
4. Draw \( \xi^{(i+1)} \sim p(\xi^{(i+1)} | Z_{ij}, \theta^{(i+1)}, \gamma^{(i+1)}) \);
5. Draw \( \gamma^{(i+1)} \sim p(\gamma^{(i+1)} | W_{ij}, \mu_a^{(i+1)}, \Sigma^{(i+1)}) \);
6. Draw \( \mu_a^{(i+1)} \sim p(\mu_a^{(i+1)} | Z_{ij}, \Sigma^{(i+1)}) \);
7. Draw \( \Sigma^{(i+1)} \sim p(\Sigma^{(i+1)} | p(\xi^{(i+1)})) \).

This iterative procedure produces a sequence of \( (\theta^{(i)}, \xi^{(i)}, \gamma^{(i)}, \mu_a^{(i)}, \Sigma^{(i)}) \), \( i = 1, ... , L \). 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 $\xi$, $\gamma$ and person parameters $\theta$. As with standard Monte Carlo, with large enough samples, the posterior means of $\xi$, $\gamma$ and $\theta$ 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 $\theta_1^{(0)}=0$, $\alpha_1^{(0)}=1$, $\beta_1^{(0)}=1$, $\gamma_1^{(0)}=2$, [16], and $\mu_1^{(0)}=\mu_2^{(0)}=0$, $\sigma_1^{(0)}=\sigma_2^{(0)}=1$. The subroutine then iteratively draws random samples for $W$, $Z$, $\theta$, $\xi$, and $\gamma$ from their respective full conditional distributions specified in (5) through (9) with $\mu=0$, $\sigma=1$, and $s=5$, $t=17$. Samples associated with the hyperparameters for $\xi$ are simulated from either (10) through (13), where uniform noninformative priors are assumed for $\mu$ and $\Sigma$, or from (14) through (17), where conjugate priors are adopted for them with $\tau_0=\varphi_0=100$, $q = \varphi = 2$, and $s_0 = \varphi_0 = .001$. 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, $\theta$, $\alpha$, $\beta$, and $\gamma$, using the batching scheme described above.

### Table 1. Posterior estimates and Monte Carlos standard errors (MCSEs) for $\alpha$ with noninformative and conjugate priors assumed for $\mu$ and $\Sigma$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Noninformative priors</th>
<th>Conjugate priors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Estimate</td>
<td>MCSE</td>
</tr>
<tr>
<td>$0.0966$</td>
<td>$0.835$</td>
<td>$0.0018$</td>
</tr>
<tr>
<td>$0.0971$</td>
<td>$0.860$</td>
<td>$0.0039$</td>
</tr>
<tr>
<td>$0.4589$</td>
<td>$0.4671$</td>
<td>$0.0048$</td>
</tr>
<tr>
<td>$0.9532$</td>
<td>$0.9197$</td>
<td>$0.0136$</td>
</tr>
<tr>
<td>$0.9771$</td>
<td>$0.162$</td>
<td>$0.0044$</td>
</tr>
<tr>
<td>$0.4891$</td>
<td>$0.4459$</td>
<td>$0.0060$</td>
</tr>
<tr>
<td>$0.8599$</td>
<td>$0.803$</td>
<td>$0.0229$</td>
</tr>
<tr>
<td>$0.9427$</td>
<td>$0.857$</td>
<td>$0.0238$</td>
</tr>
<tr>
<td>$0.2727$</td>
<td>$0.3496$</td>
<td>$0.0079$</td>
</tr>
<tr>
<td>$0.6532$</td>
<td>$0.6669$</td>
<td>$0.0200$</td>
</tr>
</tbody>
</table>

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 $\alpha$, $\beta$, and $\gamma$, 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 $\mu$ and $\Sigma$. 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.

### Table 2. Posterior estimates and Monte Carlos standard errors (MCSEs) for $\beta$ with noninformative and conjugate priors assumed for $\mu$ and $\Sigma$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Noninformative priors</th>
<th>Conjugate priors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Estimate</td>
<td>MCSE</td>
</tr>
<tr>
<td>$-0.7997$</td>
<td>$-0.8640$</td>
<td>$0.0182$</td>
</tr>
<tr>
<td>$-0.5321$</td>
<td>$-0.5792$</td>
<td>$0.0237$</td>
</tr>
<tr>
<td>$0.8583$</td>
<td>$0.8638$</td>
<td>$0.0133$</td>
</tr>
<tr>
<td>$0.7237$</td>
<td>$0.7111$</td>
<td>$0.0114$</td>
</tr>
<tr>
<td>$-0.8184$</td>
<td>$-0.8571$</td>
<td>$0.0238$</td>
</tr>
<tr>
<td>$-0.5834$</td>
<td>$-0.8953$</td>
<td>$0.0151$</td>
</tr>
<tr>
<td>$0.3629$</td>
<td>$0.1315$</td>
<td>$0.0231$</td>
</tr>
<tr>
<td>$-0.9010$</td>
<td>$-0.9853$</td>
<td>$0.0238$</td>
</tr>
<tr>
<td>$-0.9339$</td>
<td>$-0.8394$</td>
<td>$0.0225$</td>
</tr>
<tr>
<td>$-0.3978$</td>
<td>$-0.4629$</td>
<td>$0.0292$</td>
</tr>
</tbody>
</table>

### Table 3. Posterior estimates and Monte Carlos standard errors (MCSEs) for $\gamma$ with noninformative and conjugate priors assumed for $\mu$ and $\Sigma$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Noninformative priors</th>
<th>Conjugate priors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Estimate</td>
<td>MCSE</td>
</tr>
<tr>
<td>$0.3497$</td>
<td>$0.2639$</td>
<td>$0.0144$</td>
</tr>
<tr>
<td>$0.2913$</td>
<td>$0.2636$</td>
<td>$0.0169$</td>
</tr>
<tr>
<td>$0.0473$</td>
<td>$0.0550$</td>
<td>$0.0035$</td>
</tr>
<tr>
<td>$0.0497$</td>
<td>$0.0511$</td>
<td>$0.0021$</td>
</tr>
<tr>
<td>$0.3113$</td>
<td>$0.2582$</td>
<td>$0.0213$</td>
</tr>
<tr>
<td>$0.4948$</td>
<td>$0.2735$</td>
<td>$0.0138$</td>
</tr>
<tr>
<td>$0.2453$</td>
<td>$0.1486$</td>
<td>$0.0105$</td>
</tr>
<tr>
<td>$0.4687$</td>
<td>$0.3708$</td>
<td>$0.0243$</td>
</tr>
<tr>
<td>$0.1720$</td>
<td>$0.2739$</td>
<td>$0.0198$</td>
</tr>
<tr>
<td>$0.3001$</td>
<td>$0.2472$</td>
<td>$0.0205$</td>
</tr>
</tbody>
</table>

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, $\mu_\xi$ or $\Sigma_\xi$. Further, the user can change the source code so that the prior distribution for $\theta$ assumes a different location $\mu$, or scale $\sigma^2$. 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 $\tau$ or $\epsilon$ to specify different prior densities for the hyperparameters $\mu_\xi$ and $\Sigma_\xi$. 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)

INTEGER L, COUNT, IRANK, Y(N,K), BURNIN, BN, PRIOR, INDEX(K), 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(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(K) REAL C(K), NP, P, CV(L,K), IR, SD, TD, AP0, AP1, GP0, GP1, GP2

DOUBLE PRECISION BB, TMP

c**********************************************************************************
c Connect to external libraries for normal c RNOR), uniform (RNUN), beta (RNBET) and c gamma (RNGAM) random number generator, c c inverse (ANORIN, DNORIN) and CDF (ANORDF, c DNORDF) for the standard normal distribution c c and Cholesky factorization (CHFAC) routines c **********************************************************************************

EXTERNAL RNOR, RNSET, RNUN, ANORDF, ANORIN, c CHFAC, DNORDF, DNORIN, RNBM, SSCAL, c RNGAM

c**********************************************************************************
c Set initial values for item parameters c c alpha(A), beta(G), gamma(C), and person c abilities theta(TH) so that alpha=1, beta=0, c gamma=.2 for all k items, and theta=0 for c all n persons 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 Set initial values for the prior means  and  c c variances for alpha and beta c **********************************************************************************

AMU = 0.0
GMU = 0.0
AVAR = 1.0
GVAR = 1.0

c**********************************************************************************
c MU and VAR are the mean and standard c c deviation for the prior distribution of c c theta c **********************************************************************************

MU = 0.0
VAR = 1.0

c**********************************************************************************
c Specify the hyperparameters in the conjugate c c prior distributions for AMU, GMU, AVAR and c c GVAR 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 Set values for the hyperparameters (SD, TD) c
SD = 5.0
TD = 17.0

COUNT = 0
DO 30 IT = 1, L
COUNT = COUNT + 1
30 CONTINUE

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

V = 1/SUM(A*A)
PVAR = 1/(1/V + 1/VAR)
DO 50 I = 1, N
DO 50 J = 1, K
MSUM = 0.0
MSUM = MSUM + A(J)*(Z(I,J) + G(J))
50 CONTINUE

MN = MSUM*V
PMEAN = (MN/V + MJ/VAR)*PVAR
CALL RNNOR(1,TT)
TH(1) = TT*SQRT(PVAR) + PMEAN
THV(1,COUNT) = TH(1)
50 CONTINUE

DO 70 J = 1, K
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

DO 100 J = 2, 2
DO 100 I = 1, N
ZV(I,1) = Z(I,J)
100 CONTINUE

DO 110 J = 1, K
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)
110 CONTINUE

DO 90 J = 1, K
DO 90 I = 1, N
ZV(I,1) = Z(I,J)
90 CONTINUE

CALL CHFAC (2, IX, 2, 0.00001, IRANK, AMAT, 2)

DO 100 J = 1, K
DO 100 I = 1, N
ZV(I,1) = Z(I,J)
100 CONTINUE

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)

CALL MIGS(AGVAR, 2, SIGMA, INDX)
XX = MATMUL(TRANSPOSE(X),X)+SIGMA
CALL MIGS(XX,2,IX,INDX)
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 RNMDT (1, PIN, QIN, TT)
C(J) = TT
CV(COUNT,J) = TT
110 CONTINUE

c*******************************************
c Calculate the batch means and mcse's for c
alpha, beta, gamma and theta and store them c
in ITEM and PERSON
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
  230 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

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

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


