Listening. Learning. Leading.®

Research Report

ETS RR-13-32

A General Program for Item-Response Analysis That Employs the Stabilized Newton-Raphson Algorithm

Shelby J. Haberman

December 2013

ETS Research Report Series

A General Program for Item-Response Analysis That Employs the Stabilized Newton-Raphson Algorithm

Shelby J. Haberman

Educational Testing Service, Princeton, New Jersey

December 2013

Action Editor: Frank Rijmen

Reviewers: Robert Mislevy and Peter van Rijn

Abstract

A general program for item-response analysis is described that uses the stabilized Newton–Raphson algorithm. This program is written to be compliant with Fortran 2003 standards and is sufficiently general to handle independent variables, multidimensional ability parameters, and matrix sampling. The ability variables may be either polytomous or multivariate normal. Items may be dichotomous or polytomous.

Key words: log-linear models, exponential families

To facilitate flexibility of model selection and to improve statistical procedures associated with item-response theory (IRT), a computer program for item-response analysis has been constructed. The program is designed to be compliant with standards for Fortran 2003. In practice, it is implemented to be successfully compiled with PGI Fortran Version 11.1 and with gfortran 4.6.1. The program is designed to treat one-parameter logistic (1PL), two-parameter logistic (2PL), three-parameter logistic (3PL), partial credit (PC), generalized partial credit (GPC), and nominal models, as well as mixed cases in which different items satisfy different models. The program can be used with both one-dimensional and multidimensional latent variables. These variables can be polytomous, or they can be continuous and have normal distributions. The program is also capable of treating bifactor models and restricted bifactor models. Weights, matrix sampling, and independent variables are permitted. The programming is based on the stabilized Newton–Raphson algorithm (Haberman, 1988) rather than on the expectation-maximization (EM) algorithm (Bock & Aitkin, 1981; Muraki, 1991). This change facilitates computation of estimated asymptotic standard deviations of parameters and thus facilitates examination of parameter identification. The program applies to IRT models that can be described in terms of linear models for natural parameters of exponential models. Section 1 discusses the data considered by the program. Section 2 describes the general class of models that can be treated. Section 3 describes the stabilized Newton–Raphson algorithm (Haberman, 1988) employed for computations and provides details concerning its properties. Section 4 provides input and output specifications.

Although the basic structure of the program is not expected to change much in the future, it should be noted that the program and documentation are expected to be updated periodically to accommodate additional procedures for model checking and additional estimates of population parameters. In addition, a graphical user interface for the program is currently being prepared. The current documentation is presented in a belief that it is better to provide documentation now for a working program than to wait a substantial period of time until a final version is produced.

Copies of the files referenced in this report can be downloaded at http://www.ets.org/research/media/Research/RR-13-32-files.zip

1 Data

Data consist of item responses, examinee weights, and independent variables. As common in IRT, each examinee receives a collection of test items. To each item corresponds a finite number of scores. Not all examinees need to receive the same items, and the items may be multiple-choice items or constructed-response items that have a finite number of possible scores. The $J > 1$ items are numbered from 1 to J, and the $n > 1$ examinees are numbered from 1 to n. If item j, $1 \le j \le J$, is presented to examinee i, then the observed item response X_{ij} has $G_j > 1$ possible integer values, and these values are the integers from 0 to $G_j - 1$. Thus $G_j = 2$ if item j is dichotomous. If item j is not presented to examinee i, then X_{ij} is some integer that is either negative or at least G_j. Thus the examination responses can be described by an n by J array. For examinee i, the set of presented items is $\mathcal{J}_i$, and the indicator χ_{ij} is 1 for j in $\mathcal{J}_i$ and 0 for an integer j such that $1 \le j \le J$, but j is not in $\mathcal{J}_i$. In simple cases in which each examinee receives all items, each $\mathcal{J}_i$ is the set of integers from 1 to J. To avoid trivial identification problems, it is always assumed that each item j, $1 \le j \le J$, is presented to some examinee i. The array $\mathcal{J}$ provides the set $\mathcal{J}_i$ for $1 \le i \le n$. For each examinee i, the notation $\mathbf{X}_i$ is used for the array of responses X_{ij}, j in $\mathcal{J}_i$, to items presented to examinee i. The notation $\mathbf{X}$ is employed for an array of individual item responses $\mathbf{X}_i$, $1 \le i \le n$. For each examinee i, the set of possible values of $\mathbf{X}_i$ is $\mathcal{X}_i$, so that $\mathcal{X}_i$ consists of arrays $\mathbf{x}$ of integers x_j, $0 \le x_j < G_j$, j in $\mathcal{J}_i$. The set of possible values of $\mathbf{X}$ is $\mathcal{X}$, so that $\mathcal{X}$ consists of arrays $\mathbf{x}$ of $\mathbf{x}_i$ in $\mathcal{X}_i$, $1 \le i \le n$.

Associated with examinee i is a sample weight $w_i > 0$. In simple cases, this sample weight w_i is 1; however, other weights are often used in assessments that are parts of population surveys. In addition, $U \ge 1$ real predicting variables are observed for each examinee. The predicting (explanatory) variables for examinee i are Z_{iu}, $1 \le u \le U$, and the U-dimensional vector $\mathbf{Z}_i$ has elements Z_{iu}, $1 \le u \le U$. The n by U matrix $\mathbf{Z}$ has row i and column u equal to Z_{iu} for $1 \le i \le n$ and $1 \le u \le U$. The convention is adopted that $Z_{i1} = 1$, so that the first independent variable is simply the constant 1. The choice of predictors varies with the application. Possible predictors may be indicators for test administrations, membership in a gender group, age of the examinee, or years of education of the examinee. Thus, if a single categorical predictor with $U > 1$ categories is considered, then Z_{iu}, $2 \le U$, may be defined as 1 if the categorical predictor has value u and 0 otherwise.

2 Model Definition

The models under study are latent-structure models. Their definition requires an initial consideration of random vectors and conditional probabilities. The basic latent-structure model may then be defined, and the specific class of latent-structure models to be studied can then be examined.

2.1 Random Vectors and Conditional Probabilities

It is assumed that the individual responses $\mathbf{X}_i$, $1 \leq i \leq n$, are random vectors that are conditionally independent given the predictors $\mathbf{Z}_i$. For some nonempty set $\mathcal{Z}$ of U-dimensional vectors, it is assumed that $\mathbf{Z}_i$ is in $\mathcal{Z}$ for each examinee i. Thus $\mathbf{Z}$ is in the set $\mathcal{Z}^n$ of n by U matrices $\mathbf{z}$ with rows $\mathbf{z}_i$ in $\mathcal{Z}$, $1 \leq i \leq n$. For a possible response $\mathbf{x}$ of examinee i in the set $\mathcal{X}_i$ of all response arrays for that examinee and for a possible predictor vector $\mathbf{z}$ in the set $\mathcal{Z}$ of all possible values of the predictor vector $\mathbf{Z}_i$, $p_i(\mathbf{x}|\mathbf{z}) > 0$ denotes the conditional probability that $\mathbf{X}_i = \mathbf{x}$ given that $\mathbf{Z}_i = \mathbf{z}$. It is convenient to consider several arrays of conditional probabilities. For examinee i, the array $\mathbf{p}_i(\cdot|\mathbf{z})$ has elements $p_i(\mathbf{x}|\mathbf{z})$ for $\mathbf{x}$ in $\mathcal{X}_i$ and $\mathbf{z}$ in $\mathcal{Z}$. For all n examinees, $\mathbf{p}(\cdot|\mathbf{z})$ has elements $\mathbf{p}_i(\mathbf{x}|\mathbf{z}_i)$ for $\mathbf{x}$ in $\mathcal{X}_i$, $1 \leq i \leq n$, and for $\mathbf{z}$ in $\mathcal{Z}^n$.

2.2 Latent-Structure Models

The probability models under study are latent-structure models. Associated with each examinee i is a $K \geq 1$-dimensional random latent vector $\boldsymbol{\theta}_i$ with elements θ_{ik}, $1 \leq k \leq K$. The set of possible values of $\boldsymbol{\theta}_i$ is Ω for each examinee i. In addition, for each presented item j, there is an underlying polytomous random variable Y_{ij} that determines the value of the observed response X_{ij}. In many cases, Y_{ij} and X_{ij} are exactly the same; however, in situations that involve models for guessing behavior or noncompensatory item-response models, the more general definition of Y_{ij} is helpful. Because Y_{ij} determines X_{ij}, the random variable Y_{ij} has $H_j \geq G_j$ possible values. These values are integers from 0 to $H_j - 1$. The relationship of Y_{ij} to X_{ij} is specified by a category mapping. For simplicity, this mapping is a nondecreasing function. Thus, for integers H_{xj}, $0 \leq x \leq G_j$, $H_{0j} = 0$, $H_{G_j j} = H_j - 1$, and $H_{xj} < H_{(x+1)j}$ for $0 \leq x < G_j - 1$. The set $\mathcal{H}_{xj}$ consists of the integers h such that $H_{xj} \leq h < H_{(x+1)j}$ for $0 \leq x < G_j - 1$. If Y_{ij} is in $\mathcal{H}_{xj}$ and $0 \leq x < G_j - 1$, then $X_{ij} = x$. In the simplest cases, Y_{ij} and X_{ij} are the same, so that $H_{xj} = x$ for

$0 \leq x < G_j$, and $G_j = H_j$; however, the more general formulation can be helpful in the treatment of models that involve guessing or in the case of noncompensatory models.

In the case of guessing, one might have $Y_{ij} = 2Y_{ij1} + Y_{ij2}$, where Y_{ij1} and Y_{ij2} have values 0 or 1. Thus $H_j = 4$. If $G_j = 2$, $H_{0j} = 0$, $H_{2j} = 3$, and $H_{1j} = 1$, then X_{ij} is 0 if Y_{ij1} and Y_{ij2} are both 0. Here Y_{ij1} is 1 if the examinee knows the right response, whereas Y_{ij2} is 1 if the examinee correctly guesses the right response. Otherwise X_{ij} is 1.

In a noncompensatory model in which a correct response requires that three conditions must be met to provide a correct solution, one might have $Y_{ij} = 4Y_{ij1} + 2Y_{ij2} + Y_{ij3}$, where Y_{ij1}, Y_{ij2}, and Y_{ij3} have values 0 or 1. Thus $H_j = 8$. If $G_j = 2$, $H_{0j} = 0$, $H_{2j} = 7$, and $H_{1j} = 6$, then X_{ij} is 1 if, and only if, Y_{ij1}, Y_{ij2}, and Y_{ij3} are all 1.

Several arrays are often employed to treat the relationships between X_{ij} and Y_{ij} for j in $\mathcal{J}_i$. The array $\mathbf{H}$ has elements H_{xj}, $0 \leq x \leq H_j$, $1 \leq j \leq J$. The random array $\mathbf{Y}_i$ has elements Y_{ij}, j in $\mathcal{J}_i$.

The latent-structure assumptions are made that the pairs $(\mathbf{Y}_i, \boldsymbol{\theta}_i)$, $1 \leq i \leq n$, are conditionally independent given the predictor array $\mathbf{Z}$, and, for each examinee i, the local-independence assumption is made that the Y_{ij}, j in $\mathcal{J}_i$, are conditionally independent given $\boldsymbol{\theta}_i$ and $\mathbf{Z}_i$. Conditional distributions of individual item responses are assumed to be consistent for all examinees in the following sense. For each item j, $1 \leq j \leq J$, each nonnegative integer $y < H_j$, each $\mathbf{z}$ in $\mathcal{Z}$, each $\boldsymbol{\omega}$ in Ω, and each examinee i for which item j is presented (j is in $\mathcal{J}_i$), a positive real number $p_{Yj}(y|\boldsymbol{\omega})$ is the conditional probability that $Y_{ij} = y$ given that $\boldsymbol{\theta}_i = \boldsymbol{\omega}$ and $\mathbf{Z}_i = \mathbf{z}$. Thus, for each examinee i, $1 \leq i \leq n$, item j in $\mathcal{J}_i$, nonnegative integer $x < G_j$, vector $\mathbf{z}$ in $\mathcal{Z}$, and vector $\boldsymbol{\omega}$ in Ω, the conditional probability that $X_{ij} = x$ given that $\boldsymbol{\theta}_i = \boldsymbol{\omega}$ and $\mathbf{Z}_i = \mathbf{z}$ is

$$p_j(x|\boldsymbol{\omega}) = \sum_{y \in \mathcal{H}_{xj}} p_{Yj}(y|\boldsymbol{\omega}). \tag{1}$$

The conditional probability that $Y_{ij} = y$, $0 \leq y < H_j$, given that $X_{ij} = x$, $\boldsymbol{\theta}_i = \boldsymbol{\omega}$, and $\mathbf{Z}_i = \mathbf{z}$, is then

$$p_j(y|x, \boldsymbol{\omega}) = \begin{cases} p_{Xj}(x|\boldsymbol{\omega})/p_{Yj}(y|\boldsymbol{\omega}), & y \in H_{xj}, \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

The latent-structure assumptions imply that the conditional probability that $\mathbf{X}_i = \mathbf{x}$ in $\mathcal{X}_i$ given $\boldsymbol{\theta}_i = \boldsymbol{\omega}$ and $\mathbf{Z}_i = \mathbf{z}$ is then

$$p_i(\mathbf{x}|\boldsymbol{\omega}) = \prod_{j \in \mathcal{J}_i} p_j(x_j|\boldsymbol{\omega}). \tag{3}$$

The added assumption is made that, for examinee i, the conditional distribution of the latent vector $\boldsymbol{\theta}_i$ given the complete array $\mathbf{Z}$ of predictors for all examinees only depends on the prediction vector $\mathbf{Z}_i$ for examinee i. In other words, for each $\mathbf{z}_i$ in $\mathcal{Z}$, the conditional distribution of θ_i given $\mathbf{Z}_i = \mathbf{z}_i$ is the same as the conditional distribution of θ_i given $\mathbf{Z} = \mathbf{z}$ if $\mathbf{z}$ is an n by U matrix in $\mathcal{Z}^n$ with row i equal to $\mathbf{z}_i$. To describe the conditional distribution of $\boldsymbol{\theta}_i$, let $P_{\boldsymbol{\theta}}(\cdot|\mathbf{z})$ be the conditional distribution of $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i = \mathbf{z}$ in $\mathcal{Z}$. In all cases under study, the conditional moments of $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i = \mathbf{z}$ are defined for all $\mathbf{z}$ in $\mathcal{Z}$. The notation $\boldsymbol{\mu}(\mathbf{z})$ is used for the conditional expectation of $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i = \mathbf{z}$. For $1 \le k \le K$, element k of $\boldsymbol{\mu}(\mathbf{z})$ is $\mu_k(\mathbf{z})$. The notation $\boldsymbol{\Sigma}(\mathbf{z})$ is used for the conditional covariance matrix of $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i = \mathbf{z}$. For $1 \le k \le K$ and $1 \le k' \le K$, row k and column k' of $\boldsymbol{\Sigma}(\mathbf{z})$ is denoted by $\Sigma_{kk'}(\mathbf{z})$. In addition, it is helpful in the study of Hessian matrices to let $\mu_{k_1k_2k_33}(\mathbf{z})$ denote the conditional covariance of θ_{ik_1} and $\theta_{ik_2}\theta_{ik_3}$ given $\mathbf{Z}_i = \mathbf{z}$ and to let $\mu_{k_1k_2k_3k_44}(\mathbf{z})$ denote the conditional covariance of $\theta_{ik_1}\theta_{ik_2}$ and $\theta_{ik_3}\theta_{ik_4}$ given $\mathbf{Z}_i = \mathbf{z}$ for $1 \le k_e \le K$ for $1 \le e \le 4$.

Under the model, the general notation

$$p_i(\mathbf{x}|\mathbf{z}) = E(p_i(\mathbf{x}|\boldsymbol{\theta}_i)) = \int p_i(\mathbf{x}|\boldsymbol{\omega}) dP_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z}) \tag{4}$$

may be used for $\mathbf{x}$ in $\mathcal{X}_i$ for the conditional probability that $\mathbf{X}_i = \mathbf{x}$ given that $\mathbf{Z}_i = \mathbf{z}$. If Ω is finite, then $p_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z})$ denotes the conditional probability that $\boldsymbol{\theta}_i = \boldsymbol{\omega}$ in Ω given that $\mathbf{Z}_i = \mathbf{z}$, and

$$p_{\mathbf{X}|\mathbf{Z}i}(\mathbf{x}|\mathbf{z}) = \sum_{\boldsymbol{\omega}\in\Omega} p_i(\mathbf{x}|\boldsymbol{\omega}) p_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z}). \tag{5}$$

By Bayes's theorem, the conditional probability that $\boldsymbol{\theta}_i = \boldsymbol{\omega}$ in Ω given that $\mathbf{X}_i = \mathbf{x}$ and $\mathbf{Z}_i = \mathbf{z}$ is

$$p_{\boldsymbol{\theta}|\mathbf{X}\mathbf{Z}i}(\boldsymbol{\omega}|\mathbf{x},\mathbf{z}) = \frac{p_i(\mathbf{x}|\boldsymbol{\omega}) p_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z})}{p_{\mathbf{X}|\mathbf{Z}i}(\mathbf{x}|\mathbf{z})}. \tag{6}$$

If Ω is the space R^K of K-dimensional vectors and if the conditional density function of $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i = \mathbf{z}$ at $\boldsymbol{\omega}$ in Ω is $f_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z})$, then

$$p_{\mathbf{X}|\mathbf{Z}i}(\mathbf{x}|\mathbf{z}) = \int p_i(\mathbf{x}|\boldsymbol{\omega}) f_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z}) d\boldsymbol{\omega}. \tag{7}$$

By Bayes's theorem, the conditional density of $\boldsymbol{\theta}_i$ given $\mathbf{X}_i = \mathbf{x}$ and $\mathbf{Z}_i = \mathbf{z}$ is

$$f_{\boldsymbol{\theta}|\mathbf{X}\mathbf{Z}i}(\boldsymbol{\omega}|\mathbf{x},\mathbf{z}) = \frac{p_i(\mathbf{x}|\boldsymbol{\omega}) f_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z})}{p_{\mathbf{X}|\mathbf{Z}i}(\mathbf{x}|\mathbf{z})}. \tag{8}$$

The models used in the program involve a log-linear model for the conditional distribution of Y_{ij} given $\boldsymbol{\theta}_i$, item j in the set $\mathcal{J}_i$ of presented items for examinee i, and an exponential family for the conditional distribution of $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i$. These models apply to GPC models, 1PL models, 2PL models, 3PL models, and nominal-response models but not to normal ogive models.

2.3 Model Definition for Items

The log-linear model for items is expressed in terms of the relationship of Y_{ij} to $\boldsymbol{\theta}_i$, where item j is presented to examinee i. For a known positive integer D, the model involves unknown location parameters τ_{yj} for $0 \le y < H_j$ and unknown scale parameters a_{dyj}, $1 \le d \le D$, $1 \le y \le H_j$. In addition, the model involves a known D by K real matrix $\mathbf{A}$ with rows $\mathbf{A}_d$, $1 \le d \le D$, and elements A_{dk}, $1 \le d \le D$, $1 \le k \le K$. In the simplest cases, $D = K$ and $\mathbf{A}$ is the K by K identity matrix; however, models with more complex structure are often encountered.

The conditional distribution of Y_{ij} given $\boldsymbol{\theta}_i$ depends only on $\mathbf{A}\boldsymbol{\theta}_i$. To facilitate model definition, let $\mathbf{a}_{yj}$ be the D-dimensional vector with elements a_{dyj}, $1 \le d \le D$, and let

$$\mathbf{v}'\mathbf{z} = \sum_{d=1}^{D} v_d z_d$$

for any D-dimensional vectors $\mathbf{v}$ and $\mathbf{z}$ with respective elements v_d and z_d, $1 \le d \le D$. It is assumed that

$$p_{Yj}(y|\boldsymbol{\omega}) = [M_j(\boldsymbol{\omega})]^{-1} \exp(\tau_{yj} + \mathbf{a}'_{yj}\boldsymbol{A}\boldsymbol{\omega}), \tag{9}$$

where

$$M_j(\boldsymbol{\omega}) = \sum_{y=0}^{H_j-1} \exp(\tau_{yj} + \mathbf{a}'_{yj}\boldsymbol{A}\boldsymbol{\omega}). \tag{10}$$

Linear models are then applied to the parameters τ_{yj} and a_{dyj}. These models can often be expressed in terms of the logits

$$\log[p_{Yj}(y|\boldsymbol{\omega})/p_{Yj}(y'|\boldsymbol{\omega})] = \tau_{yj} - \tau_{y'j} + (\mathbf{a}_{yj} - \mathbf{a}_{y'j})'\boldsymbol{A}\boldsymbol{\omega}. \tag{11}$$

2.4 Model Definition for Latent Vectors

The latent vectors are assumed to have a probability distribution from an exponential family. A polytomous case and a normal case are considered. For both cases, the basic parameters are a K by U matrix $\boldsymbol{\Psi}$ with elements ψ_{ku}, $1 \le k \le K$, and $1 \le u \le U$, and an array $\boldsymbol{\lambda}$ with elements $\lambda_{kk'u}$, $1 \leq k \leq k' \leq K$, and $1 \leq u \leq U$. For any $\mathbf{z}$ in $\mathcal{Z}$, let $\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})$ be the K by K matrix with elements

$$\Lambda_{kk'}(\boldsymbol{\lambda}, \mathbf{z}) = \begin{cases} (1/2)\sum_{u=1}^{U} \lambda_{kk'u} z_u, & k < k', \\ \sum_{u=1}^{U} \lambda_{kk'u} z_u, & k = k', \\ (1/2)\sum_{u=1}^{U} \lambda_{k'ku} z_u, & k > k', \end{cases} \tag{12}$$

for $1 \leq k \leq K$ and $1 \leq k' \leq K$. The absolute value of the determinant of a K by K matrix $\mathbf{\Delta}$ is denoted by $|\mathbf{\Delta}|$.

2.4.1 The multivariate normal case

In the multivariate normal case, $-\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})$ is positive definite for all $\mathbf{z}$ in $\mathcal{Z}$ and the density

$$f_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z}) = (2\pi)^{-K/2}|-2\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})|^{1/2} \exp[(1/4)\mathbf{Z}'\mathbf{\Psi}'[\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})]^{-1}\mathbf{\Psi}\mathbf{z} + \boldsymbol{\omega}'\mathbf{\Psi}\mathbf{z} + \boldsymbol{\omega}'\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})\boldsymbol{\omega}], \tag{13}$$

for $\boldsymbol{\omega}$ in Ω, so that the conditional distribution of $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i = \mathbf{z}$ is multivariate normal with mean

$$\boldsymbol{\mu}(\mathbf{z}) = [-2\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})]^{-1}\mathbf{\Psi}\mathbf{z} \tag{14}$$

and covariance matrix

$$\mathbf{\Sigma}(\mathbf{z}) = [-2\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})]^{-1}. \tag{15}$$

For later reference, the function W on Ω is defined so that $W(\boldsymbol{\omega}) = 1$ for all K-dimensional vectors $\boldsymbol{\omega}$. In the multivariate normal case, $\mu_{k_1k_2k_33}(\mathbf{z}) = 0$ and for $1 \leq k_e \leq K$ for $1 \leq e \leq 3$. In addition,

$$\mu_{k_1k_2k_3k_4}(\mathbf{z}) = \Sigma_{k_1k_3}(\mathbf{z})\Sigma_{k_2k_4}(\mathbf{z}) + \Sigma_{k_1k_4}(\mathbf{z})\Sigma_{k_2k_3}(\mathbf{z}) \tag{16}$$

for $1 \leq k_e \leq K$ for $1 \leq e \leq 4$ (Isserlis, 1918).

2.4.2 The polytomous case

In the polytomous case, Ω is finite and, for some positive real numbers $W(\boldsymbol{\omega})$, $\boldsymbol{\omega}$ in Ω,

$$p_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{z}) = \frac{W(\boldsymbol{\omega})\exp[\boldsymbol{\omega}'\mathbf{\Psi}\mathbf{z} + \boldsymbol{\omega}'\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})\boldsymbol{\omega}]}{\sum_{\boldsymbol{\theta}\in\Omega} W(\boldsymbol{\theta})\exp[\boldsymbol{\theta}'\mathbf{\Psi}\mathbf{z} + \boldsymbol{\theta}'\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})\boldsymbol{\theta}]}. \tag{17}$$

In the polytomous case, no restrictions need be made on $\mathbf{\Lambda}(\boldsymbol{\lambda}, \mathbf{z})$. The random vector $\boldsymbol{\theta}_i$ satisfies a quadratic log-linear model (Haberman, 1979, chapter 6). This case is considered within the general diagnostic model (von Davier, 2008).

2.5 The Linear Model

To define the linear model, a vector $\boldsymbol{\beta}$ of dimension

$$B = (D+1)\sum_{j=1}^{J} H_j + \frac{(K(K+3)U}{2} \tag{18}$$

is defined based on the parameters τ_{hj}, a_{dhj}, ψ_{ku}, and $\lambda_{kk'u}$. Although most casual users of the program need not be concerned with this vector, knowledge of the vector definition is required for specialized applications. Let

$$\begin{aligned}
b_\tau(0,1) &= 1, & (19)\\
b_\tau(0,j) &= b_\tau(0,j-1) + H_{j-1},\ 1 < j \le J, & (20)\\
b_\tau(h,j) &= b_\tau(0,j) + h,\ 0 < h < H_j, & (21)\\
b_a(1,0,1) &= b_\tau(H_j - 1, J) + 1, & (22)\\
b_a(1,0,j) &= b_a(1,0,j-1) + DH_{j-1},\ 1 < j \le J, & (23)\\
b_a(d,0,j) &= b_a(1,0,j) + d - 1,\ 1 < d \le D, & (24)\\
b_a(d,h,j) &= b_a(d,0,j) + Dh,\ 0 < h < H_j, & (25)\\
b_\psi(1,1) &= b(D, H_J - 1, J, a) + 1, & (26)\\
b_\psi(k,1) &= b_\psi(1,1) + k - 1,\ 1 < k \le K, & (27)\\
b_\psi(k,u) &= b_\psi(k,1) + K(u-1),\ 1 < u \le U, & (28)\\
b_\lambda(1,1,1) &= b_\psi(K,U) + 1, & (29)\\
b_\lambda(1,k',1) &= b_\lambda(1,1,1) + k'(k'-1)/2,\ 1 < k' \le K, & (30)\\
b_\lambda(k,k',1) &= b_\lambda(1,k',1) + k - 1,\ 1 < k \le k', & (31)\\
b_\lambda(k,k',u) &= b_\lambda(k,k',1) + (u-1)K(K+1)/2,\ 1 < u \le U. & (32)
\end{aligned}$$

Then τ_{hj} is element $b_\tau(h,j)$ of $\boldsymbol{\beta}$, a_{dhj} is element $b_a(d,h,j)$ of $\boldsymbol{\beta}$, ψ_{ku} is element $b_\psi(k,u)$ of $\boldsymbol{\beta}$, and $\lambda_{kk'u}$ is element $b_\lambda(k,k',u)$ of $\boldsymbol{\beta}$. The linear model is defined by use of a known offset vector $\mathbf{o}$ of dimension B with elements o_b, $1 \le b \le B$, and an integer $C \ge 0$. If $C = 0$, then $\boldsymbol{\beta} = \mathbf{o}$. If $C > 0$, then the model uses a known B by C matrix $\mathbf{T}$ with row b and column c equal to T_{bc}, $1 \le b \le B$, $1 \le c \le C$, where $C \ge 1$. It is assumed in the model that

$$\boldsymbol{\beta} = \mathbf{o} + \mathbf{T}\boldsymbol{\gamma} \tag{33}$$

for some $\boldsymbol{\gamma}$ in a nonempty open subset Γ of the space R^C of C-dimensional vectors.

Given the array $\mathbf{H}$ of integers, the real matrix $\mathbf{A}$, the array of sets $\mathcal{J}$, and the weight function W on the set Ω of possible values of the $\boldsymbol{\theta}_i$, the vector $\mathbf{o}$, the matrix $\mathbf{T}$, and the space Γ, one may define the set $\mathcal{P}(\mathbf{H},\mathbf{A},\mathcal{J},W,\mathbf{o},\mathbf{T},\Gamma|\mathbf{z})$, $\mathbf{z}$ in $\mathcal{Z}^n$, to be the collection of arrays $\mathbf{p}(\cdot|\mathbf{z})$ such that (1) to (33) are satisfied for some C-dimensional vector $\boldsymbol{\gamma}$ in Γ. The model is identified for $\mathbf{Z}$ if to any member $\mathbf{p}(\cdot|\mathbf{z})$ of $\mathcal{P}(\mathbf{H},\mathbf{A},\mathcal{J},W,\mathbf{o},\mathbf{T}|\mathbf{Z})$ corresponds a unique $\boldsymbol{\gamma}$ in Γ such that (1) to (33) hold.

In many cases, added linear restrictions are imposed on $\boldsymbol{\gamma}$ to identify this parameter vector. Let $V \geq 0$ be a positive integer. If $V = 0$, then no added restrictions are used, and the set Γ_V is defined to be Γ. If $V > 0$, then a V by C matrix $\mathbf{S}$ with elements S_{vc}, $1 \leq v \leq V$, $1 \leq c \leq C$, and a V-dimensional vector $\mathbf{s}$ with elements s_v, $1 \leq v \leq V$, are given, and $\boldsymbol{\gamma}$ is required to satisfy the constraint $\mathbf{S}\boldsymbol{\gamma} = \mathbf{s}$. The set of $\boldsymbol{\gamma}$ in Γ such that $\mathbf{S}\boldsymbol{\gamma} = \mathbf{s}$ is denoted by Γ_V. It is assumed that Γ_V is not empty. The constraint normally is selected so that to any member of $\mathcal{P}(\mathbf{H},\mathbf{A},\mathcal{J},W,\mathbf{o},\mathbf{T}|\mathbf{Z})$ corresponds a unique $\boldsymbol{\gamma}$ in Γ such that $\mathbf{S}\boldsymbol{\gamma} = \mathbf{s}$.

It should be emphasized that numerous different selections of $\mathbf{H}$, $\mathbf{A}$, W, $\mathbf{o}$, and $\mathbf{T}$ can lead to the same set $\mathcal{P}(\mathbf{H},\mathbf{A},\mathcal{J},W,\mathbf{o},\mathbf{T}|\mathbf{Z})$. For example, if $\mathbf{M}$ is a nonsingular C by C matrix, then $\mathbf{TM}(\mathbf{M}^{-1}\boldsymbol{\gamma}) = \mathbf{T}\boldsymbol{\gamma}$ in (33), so that $\mathcal{P}(\mathbf{H},\mathbf{A},\mathcal{J},W,\mathbf{o},\mathbf{T}|\mathbf{Z})$ is the same as $\mathcal{P}(\mathbf{H},\mathbf{A},\mathcal{J},W,\mathbf{o},\mathbf{TM}|\mathbf{Z})$.

In practice, the matrix $\mathbf{T}$ and the offset vector $\mathbf{o}$ have decompositions $\mathbf{T} = \mathbf{T}^1\mathbf{T}^2$ and $\mathbf{o} = \mathbf{T}^1\mathbf{o}^2$, where $\mathbf{T}^1$ is a B by B^1 matrix with elements $T^1_{bb'}$, $\mathbf{T}^2$ is a B^1 by C matrix with elements T^2_{bc}, and $\mathbf{o}^2$ is a vector of dimension B^1 with elements o^2_b. Thus $\boldsymbol{\beta} = \mathbf{T}^1\boldsymbol{\beta}^1$, where $\boldsymbol{\beta}^1 = \mathbf{o}^2 + \mathbf{T}^2\boldsymbol{\gamma}$. In advanced applications, users of the program may need to make explicit use of these matrices and vectors; however, in typical cases, they are automatically constructed given basic model specifications. The matrix $\mathbf{T}^1$ and the vector $\boldsymbol{\beta}^1$ involve decompositions based on individual items. For each positive integer $j \leq J$, let $H_{\tau j}$ and H_{adj}, $1 \leq d \leq D$, be nonnegative integers less than H_j. If $H_{\tau j} > 0$, let $\mathbf{T}_{\tau j}$ be an H_j by $H_{\tau j}$ matrix. If $H_{adj} > 0$, let $\mathbf{T}_{adj}$ be an H_j by H_{adj} matrix.

Let H_{aj} be the sum of the H_{adj} for $1 \le d \le D$. Let

$$\begin{aligned}
b^1_\tau(1) &= 1, && (34)\\
b^1_\tau(j) &= b^1_\tau(j-1) + H_{\tau(j-1)},\ 1 < j \le J, && (35)\\
b^1_a(1,1) &= b^1_\tau(J) + H_{\tau J}, && (36)\\
b^1_a(1,j) &= b^1_a(1,j-1) + H_{a(j-1)},\ 1 < j \le J, && (37)\\
b^1_a(d,j) &= b^1_a(d-1,j) + H_{a(d-1)j},\ 1 < d \le D, && (38)\\
b^1_\psi &= b^1_a(1,J) + H_{aJ}. && (39)
\end{aligned}$$

Then $T^1_{bb'}$ has the following value for $0 \le h < H_j$, $1 \le h_{\tau j} \le H_{\tau j}$, $1 \le h_{adj} \le H_{adj}$, $1 \le d \le D$, $1 \le j \le J$:

1. Row $h+1$ and column $h_{\tau j}$ of $\mathbf{T}_{\tau j}$ if $b = b_\tau(h,j)$ and $b' = b^1_\tau(j) + h_{\tau j} - 1$.
2. Row $h+1$ and column h_{adj} of $\mathbf{T}_{adj}$ if $b = b_a(d,h,j)$ and $b' = b^1_a(d,j) + h_{adj} - 1$.
3. 1 if $b = b' + b_\psi(1,1) - b^1_\psi$ and $b' \ge b^1_\psi$.
4. 0 otherwise.

Let $\boldsymbol{\tau}_j$ denote the vector with elements τ_{hj} for h from 0 to $H_j - 1$. If $H_{\tau j} > 0$, let $\boldsymbol{\beta}^1_{\tau j}$ denote the vector with elements β^1_b for b from $b^1_\tau(j)$ to $b^1_\tau(j) + H_{\tau j} - 1$. Then $\boldsymbol{\tau}_j$ is $\mathbf{T}_{\tau j}\boldsymbol{\beta}^1_{\tau j}$. If $H_{\tau j}$ is 0, then $\boldsymbol{\tau}_j$ is the 0 vector of dimension H_j.

Let $\mathbf{a}^1_{dj}$ be the H_j-dimensional vector with elements a_{dhj} for $0 \le h \le H_j - 1$. If $H_{adj} > 0$, let $\boldsymbol{\beta}^1_{adj}$ denote the vector with elements β^1_b for b from $b^1_a(d,j)$ to $b^1_a(d,j) + H_{adj} - 1$. Then $\mathbf{a}^1_{dj}$ is $\mathbf{T}_{adj}\boldsymbol{\beta}^1_{adj}$. If H_{adj} is 0, then $\mathbf{a}^1_{dj}$ is the 0 vector of dimension H_j.

There are three standard cases considered in the program for $\mathbf{H}_{\tau j}$ and $\mathbf{H}_{adj}$. In the following descriptions, $D(j)$ denotes a subset of the integers 1 to D that represents elements of $\mathbf{A}\boldsymbol{\theta}_i$ related to item j. For example, in a between-item model (Adams, Wilson, & Wang, 1997), each $D(j)$ has one element, so that each item response is only related to one skill.

GPC model. The GPC model (Muraki, 1992) for Item j has $G_j = H_j$, $H_{\tau j} = H_j - 1$, and $\mathbf{T}_{\tau j}$ the cumulative matrix with row h and column $h_{\tau j}$ equal to 0 if $h \le h_{\tau j}$ and equal to 1 otherwise. If d is in $D(j)$, then $H_{adj} = 1$ and row h of $\mathbf{T}_{adj}$ has the single element $h - 1$. If d is not in $D(j)$, then $H_{adj} = 0$. If $G_j = H_j = 2$, then one has a 2PL model for the item

(Birnbaum, 1968). Note that for each h, $\log[p_j(h|\boldsymbol{\omega})/p_j(h-1|\boldsymbol{\omega})$ is an affine function of $\mathbf{A}\boldsymbol{\omega}$ with item intercept $\tau_{hj} - \tau_{(h-1)j}$ equal to element $b^1_\tau(j) + h - 1$ of $\boldsymbol{\beta}^1$ and with item slope (item discrimination) $a_{dhj} - a_{d(h-1)j}$ for element d of $\mathbf{A}\boldsymbol{\omega}$ equal to element $b^1_a(d, 1, j)$ of $\boldsymbol{\beta}^1$ if d is in $D(j)$ and equal to 0 if d is not in $D(j)$. The fundamental feature here is that the slopes do not depend on the value of h.

Nominal model. In a nominal model for item j (Bock, 1972), $H_{\tau j}$ and $\mathbf{T}_{\tau j}$ are defined as in the GPC model. In addition, if skill d is related to the item, then $H_{adj} = H_j - 1$ and $\mathbf{T}_{adj} = \mathbf{T}_{\tau j}$. If skill d is not related to the item, then $H_{adj} = 0$. This case is the same as the 2PL model if $H_j = 2$. For each h, $\log[p_j(h|\boldsymbol{\omega})/p_j(h-1|\boldsymbol{\omega})$ is an affine function of $\mathbf{A}\boldsymbol{\omega}$ with item intercept $\tau_{hj} - \tau_{(h-1)j}$ equal to element $b^1_\tau(j) + h - 1$ of $\boldsymbol{\beta}^1$ and with item slope (item discrimination) $a_{dhj} - a_{d(h-1)j}$ for element d of $\mathbf{A}\boldsymbol{\omega}$ equal to element $b^1_a(d, 1, j) + h - 1$ of $\boldsymbol{\beta}^1$ if d is in $D(j)$ and equal to 0 if d is not in $D(j)$.

3PL model. In the 3PL model (Birnbaum, 1968), $G_j = 2$, $H_j = 4$, $H_{\tau j} = 2$,

$$\mathbf{T}_{\tau j} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix},$$

$H_{adj} = 1$,

$$\mathbf{T}_{adj} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

if d is in $D(j)$, and $H_{adj} = 0$ if d is not in $D(j)$. The logit $\log[p_j(1|\boldsymbol{\omega})/p_j(0|\boldsymbol{\omega})]$ is an affine function of $\mathbf{A}\boldsymbol{\omega}$ with item intercept $\tau_{1j} - \tau_{0j}$ equal to element $b^1_\tau(j)$ of $\boldsymbol{\beta}^1$ and with item slope for element d of $\mathbf{A}\boldsymbol{\omega}$ equal to element $b^1_a(d, j)$ of $\boldsymbol{\beta}^1$ if d is in $D(j)$ and equal to 0 if d is not in $D(j)$. The logit $\log[p_j(2|\boldsymbol{\omega})/p_j(0|\boldsymbol{\omega})]$, the logit of the guessing probability, is the constant $\tau_{2j} - \tau_{0j}$ equal to element $b^1_\tau(j) + 1$ of $\boldsymbol{\beta}^1$, and the logit $\log[p_j(3|\boldsymbol{\omega})/p_j(0|\boldsymbol{\omega})]$ is the sum of the logits $\log[p_j(1|\boldsymbol{\omega})/p_j(0|\boldsymbol{\omega})]$ and $\log[p_j(3|\boldsymbol{\omega})/p_j(0|\boldsymbol{\omega})]$.

For examinee i and item j, the probability $p_j(1|\boldsymbol{\omega})$ that the observed response $X_{ij} = 1$ to item j given that the latent vector $\boldsymbol{\omega}_i = \boldsymbol{\omega}$ in Ω is then

$$p_j(1|\boldsymbol{\omega}) = \frac{\exp(\tau_{2j} - \tau_{0j})}{1 + \exp(\tau_{2j} - \tau_{0j})} + \frac{1}{1 + \exp(\tau_{2j} - \tau_{0j})} \frac{\exp[\tau_{1j} - \tau_{0j} + (\mathbf{a}_{1j} - \mathbf{a}_{0j})'\mathbf{A}\boldsymbol{\omega}]}{1 + \exp[\tau_{1j} - \tau_{0j} + (\mathbf{a}_{1j} - \mathbf{a}_{0j})'\mathbf{A}\boldsymbol{\omega}]}, \quad (40)$$

while the probability $p_j(0|\boldsymbol{\omega})$ that $X_{ij} = 0$ given that $\boldsymbol{\omega}_i = \boldsymbol{\omega}$ is then

$$p_j(0|\boldsymbol{\omega}) = \frac{1}{1 + \exp(\tau_{2j} - \tau_{0j})} \frac{1}{1 + \exp[\tau_{1j} - \tau_{0j} + (\mathbf{a}_{1j} - \mathbf{a}_{0j})'\mathbf{A}\boldsymbol{\omega}]}. \quad (41)$$

The guessing parameter is then

$$\frac{\exp(\tau_{2j} - \tau_{0j})}{1 + \exp(\tau_{2j} - \tau_{0j})}.$$

The item intercept is $\tau_{1j} - \tau_{0j}$, and the item discrimination for skill d is $a_{d1j} - a_{d0j}$.

In the Parscale program (Muraki, 1991), $D = K = 1$, $\mathbf{A}$ is the 1 by 1 identity matrix, and each item satisfies either a GPC model or a 3PL model.

The matrix $\mathbf{T}^2$ restricts the parameter vector $\boldsymbol{\beta}^1$. Some restrictions are normally required to identify parameters. The following procedure applies to most typical cases. Assume that, for each integer b from 1 to B^1, the model assumes that β_b^1 is a given constant o_b^2, assumes that $\beta_b^1 = \beta_{b'}^1$ for some $b' < b$ for which no restriction is imposed on $\beta_{b'}$, or makes no assumption about β_b^1. Let C be the number of positive integers b such that no restriction is imposed on β_b^1. Let $b(1)$ be the smallest positive integer b such that no assumption is made concerning β_b^1. For any integer c such that $1 < c \leq C$, let $b(c)$ be the smallest integer b such that $b > b(c-1)$ and no restriction is imposed on β_b^1. For $1 \leq b' \leq B^1$ and $1 \leq c \leq C$, let $T_{b'c}^2$ be 1 and $o_{b'}^2 = 0$ if $\beta_{b'}^1$ is assumed equal to $\beta_{b(c)}^1$ for $b(c) \leq b'$. Let $T_{b'c}^2 = 0$ otherwise. Let $\gamma_c = \beta_{b(c)}^1$.

Several cases are commonly encountered. Consider the following examples.

Fixed guessing. In some cases in which a 3PL model is applied to Item j, the probability of correct guessing is given. For example, it may be the inverse of the number of choices provided by a multiple-choice item. In such a case, the logit $\tau_{2j} - \tau_{0j}$ of the guessing probability is also given. This logit is element $b_\tau^1(j) + 1$ of $\boldsymbol{\beta}^2$.

Constant guessing. In some cases in which a 3PL model is applied to several items, it is assumed that the logit of the guessing probability is the same for all items. If j' is the smallest positive integer such that a 3PL model is applied to item j', then element $b_\tau^1(j) + 1$ of $\boldsymbol{\beta}^1$ is assumed equal to element $b_\tau^1(j') + 1$ of $\boldsymbol{\beta}^1$.

One-parameter models. In many cases, it is assumed that, for a given integer $d \leq D$, $a_{dhj} - a_{d(h-1)j}$ is constant for $1 \leq h \leq H_j - 1$ whenever d is in $D(j)$. This assumption is found in the 1PL model in which $H_j = G_j = 2$ for each item j (Rasch, 1960) and the 2PL model holds for each item and in the PC model (Masters, 1982) in which $H_j = G_j$ for each item and each item satisfies a GPC model. In some cases, the added restriction is made that each difference $a_{dhj} - a_{d(h-1)j}$ is 1.

No linear term for the constant. In many cases, the linear parameters ψ_{k1} corresponding to the constant predictor are assumed 0. In the case of $\boldsymbol{\theta}_i$ with an assumed normal distribution, this requirement implies that the conditional expectation of $\boldsymbol{\theta}_i$ given $Z_{iu} = 0$, $u > 1$, is the zero vector $\mathbf{0}_K$.

Fixed diagonal quadratic terms. In many cases, the quadratic parameters λ_{kk1} are assumed to be $-1/2$. If $\boldsymbol{\theta}_i$ is assumed to have a normal distribution, then the assumption made is that, for $1 \leq k \leq K$, 1 is the conditional variance of θ_{ik} given $\theta_{i'k}$, $i' \neq k$, and $Z_{iu} = 0$, $u > 1$. In this case, $o_b^2 = -1/2$ for $b = b_\lambda^1 + (k+2)(k-1)/2$, $1 \leq k \leq K$.

Fixed quadratic terms. In some cases, it is assumed that $\lambda_{kk'u} = \lambda_{kk'1}$ for $u > 1$.

Independence. It is common in the multivariate normal model for $\boldsymbol{\theta}_i$ to have the independence condition that, for some positive integer $k \leq K$, $\lambda_{kk'u} = 0$ if $k' \neq k$. Thus θ_{ik} is conditionally independent of $\theta_{ik'}$, $k' \neq k$, given $\mathbf{Z}_i$. This case also applies to models with $\boldsymbol{\theta}_i$ polytomous if Ω is a Cartesian product of finite and nonempty real sets Ω_k, $1 \leq k \leq K$.

Bifactor models. In bifactor models (Gibbons et al., 2007; Gibbons & Hedeker, 1992), $D = K > 1$, the $\boldsymbol{\theta}_i$ are assumed to have multivariate normal distributions, and the independence model applies for each integer k. The matrix $\mathbf{A}$ is the identity matrix, and each $D(j)$ has two elements, one of which is 1. Parameter identification is typically achieved by the requirement that the assumptions of fixed diagonal terms and no linear constant both apply.

Restricted bifactor models. In restricted bifactor models, a restricted variant on a bifactor model is employed with fewer parameters. Here $K = D + 1 > 2$, the $\boldsymbol{\theta}_i$ have a multivariate normal distribution, and the independence model applies for all k. The matrix $\mathbf{A}$ has elements $A_{dk} = 1$ for $d = 1$ or $d - 1 = k$ and all other elements are 0. A between-item model is assumed for the $D(j)$, $1 \leq j \leq J$. To identify parameters, one may assume that the model with no linear term for the constant applies and that λ_{111} is $-1/2$.

In one-dimensional applications with $D = K = U = 1$, $D(j)$ equals $\{1\}$ for each item j, and $\mathbf{A}$ is the 1 by 1 identity matrix, it is quite common to have the assumption that θ_{i1} has a standard normal distribution. In this case, both the case of no linear term for the constant and fixed diagonal quadratic terms apply. It follows that o_b^2 is 0 for $b = B^1 - 1$ and $-1/2$ for $b = B^1$. No γ_c corresponds to ψ_{11} or λ_{111}.

In a latent-regression model similar to the model in the National Assessment of Educational Progress, each item satisfies a GPC model or a 3PL model, the $\boldsymbol{\theta}_i$ are assumed to be multivariate normal, $U > 1$, the model with no linear term for the constant applies, the model for fixed diagonal terms applies, and the model for fixed quadratic terms applies (Mislevy, Johnson, & Muraki, 1992).

In models for concurrent calibration for $U > 1$ disjoint groups, the model for no linear term for the constant applies, and the model for fixed diagonal terms applies. The Z_{iu}, $u > 1$, are 1 for examinee i in group u and 0 otherwise.

In some cases, more general definitions of $\mathbf{T}^2$ can be employed to include other common models. For example, the linear logistic test model can be employed in this fashion (Fischer, 1973). Similarly, general latent-class models can be constructed (Heinen, 1996).

3 The Algorithm

The numerical algorithm used for maximum-marginal-likelihood estimation is a version of the stabilized Newton–Raphson algorithm (Haberman, 1988; Haberman, von Davier, & Lee, 2008) in which adaptive quadrature is employed in the multivariate normal case. To avoid trivial cases, assume that $C > 0$. Let ℓ be the weighted log-likelihood function, so that

$$\ell(\boldsymbol{\gamma}) = \ell_*(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i \ell_i(\boldsymbol{\gamma}), \tag{42}$$

where the log-likelihood component $\ell_i(\boldsymbol{\gamma}) = \ell_{i*}(\boldsymbol{\beta})$ for examinee i is the logarithm of the probability $p_i(\mathbf{X}_i|\mathbf{Z}_i)$ under equations (1) to (33). Note that ℓ_i and ℓ are functions on the space Γ of possible values of $\boldsymbol{\gamma}$, whereas ℓ_{i*} and ℓ_* are functions on the space $\mathcal{B}$ of possible values of $\boldsymbol{\beta}$. The algorithm uses the gradient $\nabla\ell_i(\boldsymbol{\gamma})$ of ℓ_i at $\boldsymbol{\gamma}$ and the Hessian matrix $\nabla^2\ell_i(\boldsymbol{\gamma})$ of ℓ_i at $\boldsymbol{\gamma}$. The gradient of ℓ at $\boldsymbol{\gamma}$ is

$$\nabla\ell(\boldsymbol{\gamma}) = \sum_{i=1}^{n} w_i \nabla\ell_i(\boldsymbol{\gamma}), \tag{43}$$

and the Hessian of ℓ at $\boldsymbol{\gamma}$ is

$$\nabla^2\ell(\boldsymbol{\gamma}) = \sum_{i=1}^{n} w_i \nabla^2\ell_i(\boldsymbol{\gamma}). \tag{44}$$

The basic Newton-Raphson algorithm uses the functions $\nabla\ell$ and $\nabla^2\ell$. In addition, the following matrix will often be employed (Louis, 1982):

$$\boldsymbol{\Phi}(\boldsymbol{\gamma}) = \sum_{i=1}^{n} w_i[\nabla\ell_i(\boldsymbol{\gamma})][\nabla\ell_i(\boldsymbol{\gamma})]', \tag{45}$$

where for C-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$ with respective elements u_c and v_c for $1 \le c \le C$, $\mathbf{uv}'$ is the C by C matrix with row c and column d equal to $u_c v_d$, $1 \le c \le C$, $1 \le d \le C$.

A value $\hat{\boldsymbol{\gamma}}$ in Γ_V is a marginal-maximum-likelihood estimate of $\boldsymbol{\gamma}$ if $\ell(\hat{\boldsymbol{\gamma}}) \ge \ell(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma}$ in Γ_V. In discussion of the algorithm, it is assumed that $\hat{\boldsymbol{\gamma}}$ is a marginal-maximum-likelihood estimate of $\boldsymbol{\gamma}$. In the ordinary Newton–Raphson algorithm for the unconstrained case of $V = 0$, an initial approximation $\boldsymbol{\gamma}_0$ to $\hat{\boldsymbol{\gamma}}$ is given, and a sequence of approximations $\boldsymbol{\gamma}_t$, $t \ge 1$, is generated by the equation

$$\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t - [\nabla^2\ell(\boldsymbol{\gamma}_t)]^{-1}\nabla\ell(\boldsymbol{\gamma}_t),\ t \ge 0. \tag{46}$$

If $\nabla^2\ell(\hat{\boldsymbol{\gamma}})$ is negative definite, then a neighborhood O of $\hat{\boldsymbol{\gamma}}$ exists such that $\boldsymbol{\gamma}_t$, $t \ge 0$, converges to $\hat{\boldsymbol{\gamma}}$ whenever $\boldsymbol{\gamma}_0$ is in Γ. In addition, if $|\mathbf{u}|$ denotes the maximum absolute value of an element of a C-dimensional vector $\mathbf{u}$, then there exists a real number $\Delta > 0$ such that, for t sufficiently large,

$$|\boldsymbol{\gamma}_{t+1} - \hat{\boldsymbol{\gamma}}| < \Delta|\boldsymbol{\gamma}_t - \hat{\boldsymbol{\gamma}}|^2 \tag{47}$$

(Kantorovich & Akilov, 1964, pp. 695–711). The constant Δ depends on the second and third differentials of ℓ.

If $V > 0$, and to any $\boldsymbol{\gamma}$ in Γ corresponds a $\boldsymbol{\gamma}_V$ in Γ_V such that $\ell(\boldsymbol{\gamma}) = \ell(\boldsymbol{\gamma}_V)$, then the Newton–Raphson algorithm remains relevant. For any real constant $\upsilon > 0$, the maximum-likelihood estimate $\hat{\boldsymbol{\gamma}}$ can be obtained by maximization of

$$\ell_V(\boldsymbol{\gamma}) = \ell(\boldsymbol{\gamma}) - \upsilon||\mathbf{S}\boldsymbol{\gamma} - \mathbf{s}||^2 \tag{48}$$

for $\boldsymbol{\gamma}$ in Γ, where, for any V-dimensional vector $\mathbf{u}$ with elements u_v, $1 \le v \le V$,

$$||\mathbf{u}|| = \sum_{v=1}^{V} u_v^2. \tag{49}$$

Let a prime be used to denote a matrix transpose. Then ℓ_V has gradient

$$\nabla\ell_V(\boldsymbol{\gamma}) = \nabla\ell(\boldsymbol{\gamma}) - 2\upsilon\mathbf{S}'(\mathbf{S}\boldsymbol{\gamma} - \mathbf{s}) \tag{50}$$

and Hessian

$$\nabla^2 \ell_V(\boldsymbol{\gamma}) = \nabla^2 \ell(\boldsymbol{\gamma}) - 2\upsilon \mathbf{S}'\mathbf{S}. \tag{51}$$

The ordinary Newton–Raphson algorithm for the constrained case uses an initial approximation $\boldsymbol{\gamma}_0$ to $\hat{\boldsymbol{\gamma}}$ such that $\boldsymbol{\gamma}_0$ is in Γ, and a sequence of approximations $\boldsymbol{\gamma}_t$, $t \geq 1$, is generated by the equation

$$\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t - [\nabla^2 \ell_V(\boldsymbol{\gamma}_t)]^{-1} \nabla \ell_V(\boldsymbol{\gamma}_t),\ t \geq 0. \tag{52}$$

If $\nabla^2 \ell_V(\hat{\boldsymbol{\gamma}})$ is negative definite, then a neighborhood O of $\hat{\boldsymbol{\gamma}}$ exists such that $\boldsymbol{\gamma}_t$, $t \geq 0$, converges to $\hat{\boldsymbol{\gamma}}$ whenever $\boldsymbol{\gamma}_0$ is in O. In addition, there exists a real number $\Delta > 0$ such that, for t sufficiently large, (47) holds. The constant Δ depends on the second and third differentials of ℓ_V. Note that ℓ_V and ℓ have the same third differentials, for they differ only by a quadratic function.

The function ℓ_V can also be employed in maximum posterior likelihood in cases in which $\boldsymbol{\gamma}$ is identified without use of any linear constraints and $\mathbf{S}$ has rank C. In this case, the corresponding prior for $\boldsymbol{\gamma}$ is normal with mean $(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\mathbf{s}$ and covariance matrix $(2\upsilon)^{-1}(\mathbf{S}'\mathbf{S})^{-1}$.

In practice, the Newton–Raphson algorithm is not readily used for typical models for item responses to estimate parameters by maximum marginal likelihood. It is difficult to obtain initial approximations $\boldsymbol{\gamma}_0$ that are sufficiently accurate to ensure convergence. As a consequence, stabilization procedures are required. These procedures can involve replacement of $\nabla^2 \ell(\boldsymbol{\gamma}_t)$ by an alternative matrix, and they can involve modification of step size. To avoid duplicate discussion, let $\ell_V = \ell$ if $V = 0$. Thus $\nabla \ell_V = \nabla \ell$ and $\nabla^2 \ell_V = \nabla^2 \ell$. In addition, let $\boldsymbol{\Phi}_V$ be $\boldsymbol{\Phi}$ if V is 0, and let $\boldsymbol{\Phi}_V$ be $\boldsymbol{\Phi} + 2\upsilon \mathbf{S}'\mathbf{S}$ if $V > 0$. The stabilized Newton–Raphson algorithm uses positive constants κ, $\tau < 2/(1 - \tau_1)$, and $\tau_1 < 1/2$. Given the initial approximation $\boldsymbol{\gamma}_0$ to $\hat{\boldsymbol{\gamma}}$, the iteration has the form

$$\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t + \alpha_t \mathbf{q}_t \ \ t \leq 0. \tag{53}$$

The vector $\mathbf{q}_t$ is $[-\nabla^2 \ell_V(\boldsymbol{\gamma}_t)]^{-1} \nabla \ell_V(\boldsymbol{\gamma}_t)$ if $\nabla^2 \ell_V(\boldsymbol{\gamma}_t)$ is negative definite. If $\nabla^2 \ell_V(\boldsymbol{\gamma}_t)$ is not negative definite and $\boldsymbol{\Phi}_V(\boldsymbol{\gamma}_t)$ is positive definite, then $\mathbf{q}_t$ is $[\boldsymbol{\Phi}_V(\boldsymbol{\gamma}_t)]^{-1} \nabla \ell_V(\boldsymbol{\gamma}_t)$. If $\nabla^2 \ell_V(\boldsymbol{\gamma}_t)$ is not negative definite and $\boldsymbol{\Phi}_V(\boldsymbol{\gamma}_t)$ is not positive definite, then $\mathbf{q}_t$ satisfies the equation

$$\boldsymbol{\Phi}_V(\boldsymbol{\gamma}_t)\mathbf{q}_t = \nabla \ell_V(\boldsymbol{\gamma}_t).$$

The constraint is imposed that if $\mathbf{u}$ is a C-dimensional vector with elements u_c, $1 \leq c \leq C$, if $\boldsymbol{\Phi}_V(\boldsymbol{\gamma}_t)\mathbf{u} = 0$, if $1 \leq d \leq D$, $u_d \neq 0$, and $u_c = 0$ for $c > d$, then element d of $\mathbf{q}_t$ is 0 (Haberman, 1979, pp. 582–585). The real number α_t is always positive. It is chosen so that $\alpha_t|\mathbf{T}\mathbf{q}_t|$ does not exceed κ and $\alpha_t \le 1$. If $\mathbf{q}_t$ is the zero vector, then $\alpha_t = 1$, $\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t$, and $\boldsymbol{\gamma}_t$ is a critical value of ℓ_V. Otherwise, α_t must satisfy the constraint that

$$\ell_V(\boldsymbol{\gamma}_{t+1}) - \ell_V(\boldsymbol{\gamma}_t) > \tau_1\alpha_t\mathbf{q}_t'\nabla\ell_V(\boldsymbol{\gamma}_t). \tag{54}$$

The value of α_t is 1 if $|\mathbf{T}\mathbf{q}_t| \le 1$ and if (54) holds for this choice of α_t.

If α_t cannot be chosen to be 1, then α_t is selected by an iterative algorithm. The initial value α_{t0} is the minimum of 1 and $\kappa/|\mathbf{T}\mathbf{q}_t|$. For any integer $\nu \ge 0$, let $\alpha_{t\nu} > 0$ be given, and let

$$\boldsymbol{\gamma}_{t\nu} = \boldsymbol{\gamma}_t + \alpha_{t\nu}\mathbf{q}_t. \tag{55}$$

If

$$\ell_V(\boldsymbol{\gamma}_{t\nu}) - \ell_V(\boldsymbol{\gamma}_t) > \tau_1\alpha_{t\nu}\mathbf{q}_t'\nabla\ell_V(\boldsymbol{\gamma}_t), \tag{56}$$

then $\alpha_{t+1} = \alpha_{t\nu}$ and $\boldsymbol{\gamma}_{t+1}$ satisfies (53). Otherwise, $\alpha_{t(\nu+1)}$ maximizes the quadratic function

$$\ell_V(\boldsymbol{\gamma}_t)) + \alpha\mathbf{q}_t'\nabla\ell_V(\boldsymbol{\gamma}_t) + (1/2)\alpha^2 c_{t\nu}$$

over real $\alpha \ge \tau\alpha_{t\nu}$, where

$$\ell(\boldsymbol{\gamma}_t)) + \alpha_{t\nu}\mathbf{q}_t'\nabla\ell(\boldsymbol{\gamma}_t) + (1/2)\alpha_{t\nu}^2 c_{t\nu} = \ell(\boldsymbol{\gamma}_{t\nu}). \tag{57}$$

Thus

$$c_{t\nu} = 2\alpha_{t\nu}^{-2}[\ell(\boldsymbol{\gamma}_{t\nu}) - \ell(\boldsymbol{\gamma}_t) - \alpha_{t\nu}\mathbf{q}_t'\nabla\ell(\boldsymbol{\gamma}_t)], \tag{58}$$

and

$$\alpha_{t(\nu+1)} = \max[\tau\alpha_{t\nu}, -c_{th}^{-1}\mathbf{q}_t'\nabla\ell(\boldsymbol{\gamma}_t)]. \tag{59}$$

The constraints on τ and τ_1 imply that

$$\tau\alpha_{t\nu} \le \alpha_{t(\nu+1)} < [2/(1-\tau_1)]\alpha_{t\nu} < \alpha_{t\nu}. \tag{60}$$

Application of Taylor's theorem shows that a positive $\nu > 0$ exists such that (56) and (55) both hold.

With the stabilized Newton–Raphson algorithm, if $\nabla^2\ell_V(\hat{\gamma})$ is negative definite, then there exists a neighborhood O_s of $\hat{\gamma}$ such that $\boldsymbol{\gamma}_t$, $t \ge 0$, converges to $\hat{\gamma}$ whenever $\boldsymbol{\gamma}_0$ is in O_s. In addition, there exists a real number $\Delta > 0$ such that, for t sufficiently large, (49) holds. The constant Δ is the same constant as in the Newton–Raphson algorithm. In addition, $\mathbf{q}_t = [-\nabla^2 \ell_V(\boldsymbol{\gamma}_t)]^{-1} \nabla \ell_V(\boldsymbol{\gamma}_t)$ and $\alpha_t = 1$ for t sufficiently large. The gain in practice is that the set O_s for the stabilized Newton–Raphson algorithm includes the set of all $\boldsymbol{\gamma}_0$ in Γ such that $\{\boldsymbol{\gamma} \in \Gamma : \ell_V(\boldsymbol{\gamma}) \geq \ell_V(\boldsymbol{\gamma}_0)\}$ is bounded and contains only one element at which the gradient of ℓ_V is the zero vector $\mathbf{0}_C$.

3.1 Formulas for Gradients

Let (1) to (33) hold. The gradient $\nabla \ell_V(\boldsymbol{\gamma})$ is evaluated from $\nabla \ell(\boldsymbol{\gamma})$ as in (50) if $V > 0$. Otherwise, $\nabla \ell_V(\boldsymbol{\gamma})$ is $\nabla \ell(\boldsymbol{\gamma})$. For evaluation of $\nabla \ell(\boldsymbol{\gamma})$, observe that $\nabla \ell(\boldsymbol{\gamma})$ is $\mathbf{T}' \nabla \ell_*(\boldsymbol{\beta})$, where $\nabla \ell_*(\boldsymbol{\beta})$ is the gradient of ℓ_* at $\boldsymbol{\beta}$ and (33) holds. In turn,

$$\nabla \ell_*(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i \nabla \ell_{i*}(\boldsymbol{\beta}), \tag{61}$$

where $\nabla \ell_{i*}(\boldsymbol{\beta})$, $1 \leq i \leq n$, is the gradient of ℓ_{i*} at $\boldsymbol{\beta}$. Recall (19) to (32). For $\boldsymbol{\omega}$ in Ω, let

$$\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) = \begin{cases} \log p_i(\mathbf{X}_i|\boldsymbol{\omega}) + \log f_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{Z}_i), & \Omega = R^K, \\ \log p_i(\mathbf{X}_i|\boldsymbol{\omega}) + \log p_{\boldsymbol{\theta}|\mathbf{Z}}(\boldsymbol{\omega}|\mathbf{Z}_i), & \Omega \text{ finite.} \end{cases} \tag{62}$$

Then element b, $1 \leq b \leq B$, of the gradient $\nabla \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega})$ is $\ell_{ib*}(\boldsymbol{\beta}|\boldsymbol{\omega})$. For $1 \leq b \leq B$, $0 \leq h \leq H_j$, $1 \leq j \leq J$, $1 \leq d \leq D$, $1 \leq k \leq K$, $1 \leq k' \leq K$, $1 \leq u \leq U$, $x = X_{ij}$, and $\mathbf{z} = \mathbf{Z}_i$,

$$\ell_{ib*}(\boldsymbol{\beta}) = \begin{cases} \chi_{ij}[p_j(h|x, \boldsymbol{\omega}) - p_{Yj}(h|\boldsymbol{\omega})], & b = b(h, j, \tau), \\ \chi_{ij}[p_j(h|x, \boldsymbol{\omega}) - p_{Yj}(h|\boldsymbol{\omega})]\mathbf{A}'_d \boldsymbol{\omega}, & b = b(d, h, j, a), \\ z_u[\omega_k - \mu_k(\mathbf{z})], & b = b(k, u, \psi), \\ z_u[\omega_k \omega_{k'} - \mu_{kk'2*}(\mathbf{z})], & b = b(k, k', u, \lambda). \end{cases} \tag{63}$$

It follows that $\nabla \ell_{i*}(\boldsymbol{\beta})$ is the conditional expectation of $\nabla \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\theta}_i)$ given $\mathbf{X}_i$ and $\mathbf{Z}_i$. By (6), if Ω is finite, then

$$\nabla \ell_{i*}(\boldsymbol{\beta}) = \sum_{\boldsymbol{\omega} \in \Omega} p_{\boldsymbol{\theta}|\mathbf{X}\mathbf{Z}i}(\boldsymbol{\omega}|\mathbf{X}_i, \mathbf{Z}_i) \nabla \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}). \tag{64}$$

If Ω is R^K, then (8) implies that

$$\nabla \ell_{i*}(\boldsymbol{\beta}) = \int f_{\boldsymbol{\theta}|\mathbf{X}\mathbf{Z}i}(\boldsymbol{\omega}|\mathbf{X}_i, \mathbf{Z}_i) \nabla \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) d\boldsymbol{\omega}. \tag{65}$$

Computation of $\nabla \ell(\boldsymbol{\gamma})$ exploits the relationship

$$\nabla \ell(\boldsymbol{\gamma}) = \mathbf{T}' \sum_{i=1}^{n} w_i \nabla \ell_{i*}(\boldsymbol{\beta}). \tag{66}$$

Similarly, computation of $\mathbf{\Phi}(\boldsymbol{\gamma})$ uses the relationship

$$\mathbf{\Phi}(\boldsymbol{\gamma}) = \mathbf{T}' \left\{ \sum_{i=1}^{n} w_i \nabla \ell_{i*}(\boldsymbol{\beta}) [\nabla \ell_{i*}(\boldsymbol{\beta})]' \right\} \mathbf{T}. \tag{67}$$

Some reduction in computations can be achieved by noting that $\ell_{ib*}(\boldsymbol{\beta}|\boldsymbol{\omega})$ is not needed for a positive integer $b \leq B$ if T_{bc} is 0 for all positive integers $c \leq C$.

Note that in practice, gradient calculations needed for computation of $\boldsymbol{\gamma}_{t+1}$ from $\boldsymbol{\gamma}_t$ for $t \geq 0$ involve substitution of $\boldsymbol{\gamma}_t$ for $\boldsymbol{\gamma}$ in (1) to (19).

3.2 Formulas for Hessian Matrices

Computation of $\nabla^2 \ell_V(\boldsymbol{\gamma})$ is similar to computation of $\nabla \ell_V(\boldsymbol{\gamma})$. The Hessian $\nabla^2 \ell_V(\boldsymbol{\gamma})$ is evaluated from $\nabla^2 \ell(\boldsymbol{\gamma})$ as in (51) if $V > 0$. Otherwise $\nabla^2 \ell_V(\boldsymbol{\gamma})$ equals $\nabla^2 \ell(\boldsymbol{\gamma})$. The Hessian $\nabla^2 \ell_{i*}(\boldsymbol{\beta})$ of $\ell_{i*}(\boldsymbol{\beta})$ is evaluated by use of the gradient $\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega})$, the gradient $\nabla \ell_{i*}(\boldsymbol{\beta})$, and the Hessian $\nabla^2 \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega})$ of $\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega})$. For $1 \leq b \leq B$ and $1 \leq b' \leq B$, let $\ell_{ibb'*}(\boldsymbol{\gamma}|\boldsymbol{\omega})$ be row b and column b' of $\nabla^2 \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega})$.

For integers h and h', let $\delta_{hh'}$ be 1 for $h = h'$ and 0 for $h \neq h'$. Let

$$\Pi_{hh'j}(\boldsymbol{\omega}) = \delta_{hh'} p_j(h|x, \boldsymbol{\omega}) - p_j(h|x, \boldsymbol{\omega}) p_{j'}(h'|x, \boldsymbol{\omega}) - \delta_{hh'} p_{Yj}(h|\boldsymbol{\omega}) + p_{Yj}(h|x, \boldsymbol{\omega}) p_{Yj'}(h'|x, \boldsymbol{\omega}) \tag{68}$$

for $0 \leq h < G_j$, $0 \leq h' < G_j$, $1 \leq j \leq J$, and $\boldsymbol{\omega}$ in Ω. For $1 \leq b \leq B$, $1 \leq b' \leq B$, $0 \leq h \leq H_j$, $1 \leq j \leq J$, $0 \leq h' \leq H_{j'}$, $1 \leq j' \leq J$, $1 \leq d \leq D$, $1 \leq k \leq K$, $1 \leq k_1 \leq k$, $1 \leq d' \leq D$, $1 \leq k' \leq K$, $1 \leq k_2 \leq k'$, $1 \leq u \leq U$, $1 \leq u' \leq U$, $x = X_{ij}$, and $\mathbf{z} = \mathbf{Z}_i$,

$$\ell_{ibb'*}(\boldsymbol{\beta}) = \begin{cases} \chi_{ij} \Pi_{hh'j}(\boldsymbol{\omega}), & b = b(h, j, \tau), b' = b(h', j, \tau), j = j', \\ \chi_{ij} \Pi_{hh'j}(\boldsymbol{\omega}) \mathbf{A}'_{d'} \boldsymbol{\omega}, & b = b(h, j, \tau), b' = b(d', h', j', a), j = j', \\ \chi_{ij} \Pi_{hh'j}(\boldsymbol{\omega}) \mathbf{A}'_{d} \boldsymbol{\omega}, & b = b(d, h, j, a), b' = b(h', j', \tau), j = j', \\ \chi_{ij} \Pi_{hh'j}(\boldsymbol{\omega}) \mathbf{A}'_{d} \boldsymbol{\omega} \mathbf{A}'_{d'} \boldsymbol{\omega}, & b = b(d, h, j, a), b' = b(d', h', j', \tau), j = j', \\ -z_u z_{u'} \mu_{kk'2}(\mathbf{z}), & b = b(k, u, \psi), b' = b(k', u', \psi), \\ -z_u z_{u'} \mu_{kk_2k'3}(\mathbf{z}), & b = b(k, u, \psi), b' = b(k_2, k', u', \lambda), \\ -z_u z_{u'} \mu_{k'k_1k}(\mathbf{z}), & b = b(k_1, k, u, \lambda), b' = b(k', u', \psi), \\ -z_u z_{u'} \mu_{k_1kk'_k4}(\mathbf{z}), & b = b(k_1, k, u, \lambda), b' = b(k_2, k', u, \lambda), \\ 0, & \text{otherwise.} \end{cases} \tag{69}$$

The Hessian matrix $\nabla^2\ell_{i*}(\boldsymbol{\beta})$ is then the sum of the conditional expectation of $\nabla^2\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\theta}_i)$ given $\mathbf{X}_i$ and $\mathbf{Z}_i$ and the conditional covariance matrix of $\nabla\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\theta}_i)$ given $\mathbf{X}_i$ and $\mathbf{Z}_i$. Let

$$\mathbf{K}_i(\boldsymbol{\beta}|\boldsymbol{\omega}) = \nabla^2\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) + [\nabla\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) - \nabla\ell_{i*}(\boldsymbol{\beta})][\nabla\ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) - \nabla\ell_{i*}(\boldsymbol{\beta})]' \tag{70}$$

for $\boldsymbol{\omega}$ in Ω. If Ω is finite, then (6) implies that

$$\nabla^2\ell_{i*}(\boldsymbol{\beta}) = \sum_{\boldsymbol{\omega}\in\Omega} p_{\boldsymbol{\theta}|\mathbf{XZ}i}(\boldsymbol{\omega}|\mathbf{X}_i, \mathbf{Z}_i)\mathbf{K}_i(\boldsymbol{\beta}|\boldsymbol{\omega}). \tag{71}$$

If Ω is R^K, then (8) implies that

$$\nabla^2\ell_{i*}(\boldsymbol{\beta}) = \int f_{\boldsymbol{\theta}|\mathbf{XZ}i}(\boldsymbol{\omega}|\mathbf{X}_i, \mathbf{Z}_i)\mathbf{K}_i(\boldsymbol{\beta}|\boldsymbol{\omega})d\boldsymbol{\omega}. \tag{72}$$

Computation of $\nabla^2\ell(\boldsymbol{\gamma})$ exploits the relationship

$$\nabla^2\ell(\boldsymbol{\gamma}) = \mathbf{T}'\left[\sum_{i=1}^{n} w_i\nabla^2\ell_{i*}(\boldsymbol{\beta})\right]\mathbf{T}. \tag{73}$$

As in the case of the gradient, Hessian calculations needed for computation of $\boldsymbol{\gamma}_{t+1}$ from $\boldsymbol{\gamma}_t$ for $t \geq 0$ involve substitution of $\boldsymbol{\gamma}_t$ for $\boldsymbol{\gamma}$ in (1) to (19).

3.3 Quadrature Procedures

Whenever the multivariate normal distribution is used for the latent vector, numerical quadrature procedures are used to evaluate required integrals. For each examinee i, a finite and nonempty set $\mathcal{Q}_i$ of quadrature points in R^K is used together with a positive weight function $\mathcal{W}_i(\mathbf{q})$ defined for each $\mathbf{q}$ in $\mathcal{Q}_i$. For a finite and nonempty subset $\mathcal{Q}$ of R^K, a positive function $\mathcal{W}$ on $\mathcal{Q}$, and a real function $\mathcal{G}$ on R^K, define

$$\mathcal{I}_{\mathcal{Q}\mathcal{W}}(\mathcal{G}) = \frac{\sum_{\mathcal{U}\in\mathcal{Q}}\mathcal{W}(\mathcal{U})\mathcal{G}(\mathcal{U})}{\sum_{\mathcal{U}\in\mathcal{Q}}\mathcal{W}(\mathcal{U})}. \tag{74}$$

If $\mathcal{G}$ is a B-dimensional vector function on R^K with elements $\mathcal{G}_b$, $1 \leq b \leq B$, let $\mathcal{I}_{\mathcal{Q}\mathcal{W}}(\mathcal{G})$ be the B-dimensional vector function with elements $\mathcal{I}_{\mathcal{Q}\mathcal{W}}(\mathcal{G}_b)$ for $1 \leq b \leq b$. Similar conventions can be used for matrix-valued functions. Let

$$\mathcal{L}_i(\boldsymbol{\omega}) = \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) \tag{75}$$

for K-dimensional vectors $\boldsymbol{\omega}$. One approximates $\ell_{i*}(\boldsymbol{\beta})$ by

$$\bar{\ell}_{i*}(\boldsymbol{\beta}) = \log[\mathcal{I}_{\mathcal{Q}_i\mathcal{W}_i}(\exp(\mathcal{L}_i))]. \tag{76}$$

Let $\nabla \ell_{i*}(\boldsymbol{\beta}|\cdot))$ be the function with value $\nabla \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega})$ for $\boldsymbol{\omega}$ in Ω. The gradient $\nabla \ell_{i*}(\boldsymbol{\beta})$ is approximated by

$$\bar{\nabla} \ell_{i*}(\boldsymbol{\beta}) = \frac{\mathcal{I}_{\mathcal{Q}_i \mathcal{W}_i}(\exp(\mathcal{L}_i) \nabla \ell_{i*}(\boldsymbol{\beta}|\cdot))}{\mathcal{I}_{\mathcal{Q}_i \mathcal{W}_i}(\exp(\mathcal{L}_i))}. \tag{77}$$

For the Hessian matrix, let

$$\bar{\mathbf{K}}_i(\boldsymbol{\beta}|\boldsymbol{\omega}) = \nabla^2 \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) + [\nabla \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) - \bar{\nabla} \ell_{i*}(\boldsymbol{\beta})][\nabla \ell_{i*}(\boldsymbol{\beta}|\boldsymbol{\omega}) - \bar{\nabla} \ell_{i*}(\boldsymbol{\beta})]'. \tag{78}$$

The Hessian matrix $\nabla^2 \ell_{i*}(\boldsymbol{\beta})$ is approximated by

$$\bar{\nabla}^2 \ell_{i*}(\boldsymbol{\beta}) = \frac{\mathcal{I}_{\mathcal{Q}_i \mathcal{W}_i}(\exp(\mathcal{L}_i) \bar{\mathbf{K}}_i(\boldsymbol{\beta}|\cdot))}{\mathcal{I}_{\mathcal{Q}_i \mathcal{W}_i}(\exp(\mathcal{L}_i))}. \tag{79}$$

3.4 Adaptive Quadrature

The weight functions $\mathcal{W}_i$ and sets $\mathcal{Q}_i$ of quadrature points are permitted to depend on the examinee i to improve the efficiency of the quadrature procedures in terms of accuracy and computational speed. Thus quadrature is adaptive. The basic use of adaptive quadrature in estimation by maximum marginal likelihood has been explored by quite a number of investigators (e.g., Naylor & Smith, 1982; Haberman, 2006; Schilling & Bock, 2005). In the algorithm under study, adaptive quadrature is based on the function $\mathcal{L}_i$ for each examinee i. A K-dimensional vector $\boldsymbol{\theta}_{im}$ is employed that approximates the location of the maximum $\hat{\boldsymbol{\theta}}_{im}$ of $\mathcal{L}_{i*}(\boldsymbol{\omega})$ over all K-dimensional vectors $\boldsymbol{\omega}$. One then computes the Hessian matrix $\nabla^2 \mathcal{L}_i(\boldsymbol{\theta}_{im})$ of $\mathcal{L}_i$ at $\boldsymbol{\theta}_{im}$. The Cholesky decomposition

$$\mathbf{L}_i \mathbf{L}_i' = -\nabla^2 \mathcal{L}_i(\boldsymbol{\theta}_{im}) \tag{80}$$

is available for a unique K by K matrix $\mathbf{L}_i$ such that, for $1 \le k \le K$ and $1 \le k' \le K$, row k and column k' is 0 if $k < k'$ and row k and column k is nonnegative (Wilkinson, 1963, pp. 117–120). If $\nabla^2 \mathcal{L}_i(\boldsymbol{\theta}_{im})$ is negative definite, then row k and column k of $\mathbf{L}_i$ is positive for $1 \le k \le K$. If $G_j = H_j$ for each item j, then it follows from standard properties of log-linear models and exponential families that $\nabla^2 \mathcal{L}_i$ is a negative definite function and $\mathcal{L}_i$ is a strictly concave function (Haberman, 1973).

Let $\nabla^2 \mathcal{L}_i$ be negative definite. Let R_i be the linear function defined by

$$R_i(\boldsymbol{\omega}) = \boldsymbol{\theta}_{im} + \mathbf{L}_i^{-1} \boldsymbol{\omega} \tag{81}$$

for K-dimensional vectors $\boldsymbol{\omega}$. If the integral $\int \mathcal{G}\exp(\mathcal{L}_i)$ is defined for a real function $\mathcal{G}$ on R^K, then

$$\int \mathcal{G}\exp(\mathcal{L}_i) = |\mathbf{L}_i|^{-1}\int \mathcal{G}(R_i))\exp[\mathcal{L}_i(R_i)]. \tag{82}$$

The right-hand side of (82) has the attraction that $\exp[\mathcal{L}_i(R_i(\boldsymbol{\omega}_i))]/\phi_{KV}(\boldsymbol{\omega}_i)$ is approximately constant and the determinant of $\mathbf{L}_i$ is the product of the diagonal elements of $\mathbf{L}_i$. Multivariate cumulants can be used to obtain the more precise result that, for a homogeneous third-degree polynomial $\mathcal{P}_{i3}$ on R^K, and a positive constant $\mathcal{R}_{3i}$,

$$|\mathcal{L}_i(R_i(\boldsymbol{\omega}_i)) - \mathcal{L}_i(\boldsymbol{\theta}_{im}) + (K/2)\log(2\pi) - \log\phi_{KV}(\boldsymbol{\omega}_i - \mathcal{P}_{i3}(\boldsymbol{\omega}_i)| < \mathcal{R}_{3i}|\boldsymbol{\omega}_i|^4 \tag{83}$$

(McCullagh & Nelder, 1989, pp. 165–167). In addition, $\mathcal{G}(R_i(\boldsymbol{\omega}_i))$ is approximately constant if $\mathcal{G}$ is differentiable and if the maximum ratio $|\boldsymbol{\omega}|/|\mathbf{L}_i\boldsymbol{\omega}|$ is small for $\boldsymbol{\omega}$ a nonzero K-dimensional vector. As a consequence, evaluation of the right-hand side of (82) is readily accomplished by use of quadrature procedures designed for integrals of the form $\int \mathcal{G}\phi_{KV}$. Application of standard arguments from Bayesian inference to IRT (Holland, 1990) suggest that $\mathcal{P}_{i3}$ is typically on the order of the inverse of the square root of the number J_i of items presented to examinee i and $\mathcal{R}_{3i}$ is typically on the order of J_i^{-1}.

Determination of an approximation $\boldsymbol{\theta}_{im}$ to $\hat{\boldsymbol{\theta}}_i$ can be performed by the same stabilized Newton–Raphson algorithm already described for use in approximation of $\hat{\boldsymbol{\gamma}}$. In practice, $\boldsymbol{\gamma}$ is replaced by $\boldsymbol{\gamma}_t$ in all formulas required for computation of $\boldsymbol{\gamma}_{t+1}$.

In the program, adaptive quadrature is applied in the following fashion. A finite and nonempty set of quadrature points $\mathcal{Q}$ in R^K and a positive function $\mathcal{W}$ on $\mathcal{Q}$ are given. The set $\mathcal{Q}_i$ then consists of the vectors $R_i(\mathcal{U})$ for $\mathcal{U}$ in $\mathcal{Q}$, and

$$\mathcal{W}_i(R_i(\mathcal{U})) = |\mathbf{L}_i|^{-1}\mathcal{W}(\mathcal{U})/\phi_{KV}(\mathcal{U}) \tag{84}$$

for $\mathcal{U}$ in $\mathcal{Q}$. Typically $\mathcal{Q}$ and $\mathcal{W}$ are selected so that, for a set $\mathcal{S}$ of polynomials on R^K,

$$\int \mathcal{P}\phi_{KV} = \mathcal{I}_{\mathcal{QW}}(\mathcal{P}),\ \mathcal{P}\in\mathcal{S}. \tag{85}$$

3.5 Product Rules

In many cases, the set $\mathcal{Q}$ and the function $\mathcal{W}$ are constructed by use of product rules. For each integer k, $1 \le k \le K$, a finite and nonempty real set $\mathcal{T}_k$ and a positive real function $\mathcal{Y}_k$ on $\mathcal{T}_k$

are given. The set $\mathcal{T}_k$ has Q_k members. The set $\mathcal{Q}$ is the Cartesian product $\prod_{k=1}^{K} \mathcal{T}_k$ of the $\mathcal{T}_k$, $1 \le k \le K$, so that $\mathcal{U}$ is in $\mathcal{Q}$ if, and only if, each element $\mathcal{U}_k$ of $\mathcal{U}$, $1 \le k \le K$, is in $\mathcal{T}_k$. The set $\mathcal{Q}$ has $\prod_{k=1}^{K} Q_k$ elements. The weight function $\mathcal{W}$ on $\mathcal{Q}$ is $\bigotimes_{k=1}^{K} \mathcal{Y}_k$, so that

$$\mathcal{W}(\mathcal{U}) = \prod_{k=1}^{K} \mathcal{Y}_k(\mathcal{U}_k). \tag{86}$$

Properties of $\mathcal{I}_{\mathcal{QW}}$ are derived from the properties of the $\mathcal{I}_{\mathcal{T}_k \mathcal{Y}_k}$. If, for $1 \le k \le K$, $\mathcal{G}_k$ is a real function on the real line such that $\mathcal{G}_k \phi$ is integrable and

$$\mathcal{I}_{\mathcal{T}_k \mathcal{Y}_k}(\mathcal{G}_k) = \int \mathcal{G}_k \phi, \tag{87}$$

then $\mathcal{G} = \bigotimes_{k=1}^{K} \mathcal{G}_k$ satisfies

$$\mathcal{I}_{\mathcal{QW}}(\mathcal{G}) = \int \mathcal{G} \phi_{VK} \tag{88}$$

(Haberman, 1996, chapter 4).

The basic cases of product rules considered in the program are Gauss–Hermite quadrature and evenly spaced quadrature with normal weights; however, the program can treat other product rules as well. In addition, the program does consider alternatives to product rules in which $\mathcal{Q}$ is a nonempty proper subset of $\prod_{k=1}^{K} \mathcal{T}_k$ and a positive real weighting adjustment function $\mathcal{A}$ on $\mathcal{Q}$ is employed. The weight function $\mathcal{W}'$ on $\mathcal{Q}$ satisfies

$$\mathcal{W}'(\mathcal{U}) = \mathcal{A}(\mathcal{U})\mathcal{W}(\mathcal{U}) \tag{89}$$

for $\mathcal{U}$ in $\mathcal{Q}$. Multiplication of all $\mathcal{A}$ by the same positive constant leads to the same quadrature procedure. This result is sometimes helpful in data input.

3.6 Gauss–Hermite Quadrature

The most common selection of $\mathcal{Q}$ and $\mathcal{W}$ is based on Gauss–Hermite quadrature (Davis & Polonsky, 1965; Hochstrasser, 1965; Ralston, 1965). This approach is well known, especially in the case of one-dimensional quadrature; however, a number of variations exist in the literature concerning the exact scaling employed. In addition, many properties related to weak convergence appear to be little known.

In the Gauss–Hermite quadrature used in the program, integers $Q_k > 1$ are given for $1 \le k \le K$. The set $\mathcal{Q}$ is defined by use of Hermite polynomials defined on the real line. For an integer $Q \geq 0$, the Qth Hermite polynomial is defined so that $\mathcal{H}_0$ is the constant function 1 and, for $Q > 0$, $(-1)^Q \mathcal{H}_Q \phi$ is the Qth derivative of the standard normal density function ϕ (Cramér, 1946, p. 133). For $\mathcal{U}$ real, $\mathcal{H}_0(\mathcal{U}) = 1$, $\mathcal{H}_1(\mathcal{U}) = \mathcal{U}$, $\mathcal{H}_2(\mathcal{U}) = \mathcal{U}^2 - 1$, and

$$\mathcal{H}_{Q+1}(\mathcal{U}) = \mathcal{U}\mathcal{H}_Q(\mathcal{U}) - Q\mathcal{H}_{Q-1}(\mathcal{U}) \tag{90}$$

for $Q > 0$. For Q even, $\mathcal{H}_Q$ has the symmetry property that $\mathcal{H}_Q(\mathcal{U}) = \mathcal{H}_Q(-\mathcal{U})$ for all real $\mathcal{U}$. For Q odd, $\mathcal{H}_Q$ has the antisymmetry property that $\mathcal{H}_Q(\mathcal{U}) = -\mathcal{H}_Q(-\mathcal{U})$ for all real $\mathcal{U}$.

Corresponding to $\mathcal{H}_Q$ is a set $\mathcal{N}_Q$ of Q real numbers such that $\mathcal{H}_Q(\mathcal{U}) = 0$ for a real number $\mathcal{U}$ if, and only if, $\mathcal{U}$ is in $\mathcal{N}_Q$. The set $\mathcal{N}_Q$ has the symmetry property that $\mathcal{U}$ is in $\mathcal{N}_Q$ if, and only if, $-\mathcal{U}$ is in $\mathcal{N}_Q$.

In addition to use of standard tables (Davis & Polonsky, 1965), a number of numerical procedures can be employed to find $\mathcal{N}_Q$ (Golub & Welsch, 1969). Corresponding to $\mathcal{N}_Q$ is the weight function $\mathcal{V}_Q$ on $\mathcal{N}_Q$ such that

$$\mathcal{V}_Q(\mathcal{U}) = \frac{Q!}{Q^2[H_{Q-1}(\mathcal{U})]^2} \tag{91}$$

for $\mathcal{U}$ in $\mathcal{N}_Q$. The function $\mathcal{V}_Q$ has the symmetry property that $\mathcal{V}_Q(\mathcal{U}) = \mathcal{V}_Q(-\mathcal{U})$ for all $\mathcal{U}$ in $\mathcal{N}_Q$. If $Q = 2$, then $\mathcal{N}_Q$ is the set with elements -1 and 1, and $\mathcal{V}_Q$ is always 1/2. If $Q = 3$, then $\mathcal{N}_Q$ consists of $-3^{1/2}$, 0, and $3^{1/2}$, and $\mathcal{V}_Q(\mathcal{U})$ is 1/6 for $\mathcal{U}$ equal $-3^{1/2}$ or $3^{1/2}$ and 2/3 for $\mathcal{U}$ equal to 0.

The set $\mathcal{Q}$ for adaptive quadrature is then $\prod_{k=1}^{K} \mathcal{N}_{Q_k}$, and the weight function $\mathcal{W}$ satisfies

$$\mathcal{W} = \prod_{k=1}^{K} \mathcal{V}_{Q_k}. \tag{92}$$

Equation (85) holds if $\mathcal{S}$ consists of all polynomial functions $\mathcal{P}$ on R^K such that, for each integer k, $1 \leq k \leq K$, for element k of the argument of the polynomial, each term of $\mathcal{P}$ either is of odd degree or of degree no greater than $2Q_k - 1$. For example, consider the case of $K = 3$ and $Q_k = 4$ for $1 \leq k \leq K$. Let $\mathcal{P}$ satisfy

$$\mathcal{P}(\mathcal{U}) = 3\mathcal{U}_1\mathcal{U}_2^9\mathcal{U}_3^6 - 5\mathcal{U}_1^9\mathcal{U}_2^4$$

for $\mathcal{U}$ in R^3 with elements $\mathcal{U}_k$ for $1 \leq k \leq 3$. Then $\mathcal{P}$ is in $\mathcal{S}$.

A classical error formula is available for the one-dimensional case of $K = 1$. Let $\mathcal{G}$ be a real function on the real line which is $2Q$ times continuously differentiable. Let the $2Q$th derivative of $\mathcal{G}$ be $\mathcal{G}_{2Q}$. Then

$$\int \mathcal{G}\phi - \mathcal{I}_{\mathcal{Q}\mathcal{W}}(\mathcal{G}) = \frac{\mathcal{G}_{2Q}(\mathcal{U})Q!}{2Q)!} \tag{93}$$

for some real $\mathcal{U}$. A notable consequence of this formula is that $\int \mathcal{G}\phi \geq \mathcal{I}_{\mathcal{QW}}(\mathcal{G})$ if m is an even nonnegative integer and $\mathcal{G}(\mathcal{U}) = \mathcal{U}^m$ for all real $\mathcal{U}$.

Weak convergence results can be applied to $\mathcal{I}_{\mathcal{QW}}$ (Rao, 1973, chapter 2). It is helpful to begin with the case of $K = 1$. Let $\mathcal{G}$ be a measurable real function on the real line that is continuous almost everywhere with respect to Lebesgue measure. If, for some polynomial function $\mathcal{G}^*$ on the real line, $|\mathcal{G}| \leq \mathcal{G}^*$, then $\mathcal{G}\phi$ has a finite integral and $\mathcal{I}_{\mathcal{QW}}(\mathcal{G})$ converges to $\int \mathcal{G}\phi$ as Q_1 approaches ∞ (Haberman, 1996, chapter 4). More generally, (93) can be used to show that if, for some real function $\mathcal{G}^*$ on the real line, $|\mathcal{G}| \leq \mathcal{G}^*$, $\mathcal{G}^*$ is infinitely differentiable with Qth derivative $\mathcal{G}_Q$, and for some nonnegative integer Q^*, $\mathcal{G}_Q$ is a nonnegative function for all $Q \geq Q^*$, and $\mathcal{G}^*\phi$ has a finite integral, then $\mathcal{I}_{\mathcal{QW}}(\mathcal{G})$ converges to $\int \mathcal{G}\phi$ as Q_1 approaches ∞ (Haberman, 1996, chapter 4). For example, for any real $c > 0$ and real $d < 1/2$, this result can be applied if $\mathcal{G}^*(\mathcal{U}) = c\exp(d\mathcal{U}^2)$ for $\mathcal{U}$ real.

To obtain results concerning weak convergence for $K > 1$, one uses (87) and (86). Use of characteristic functions permits the results for $K = 1$ to be applied to verify that $\mathcal{G}\phi_{KV}$ has a finite integral and $\mathcal{I}_{\mathcal{WQ}}(\mathcal{G})$ converges to $\int \mathcal{G}\phi_{KV}$ as $\min_{1\leq k\leq K} Q_k$ approaches ∞ if $\mathcal{G}$ is a Lebesgue-measurable real function on R^K, $\mathcal{G}$ is continuous almost everywhere with respect to Lebesgue measure, and, for some nonnegative continuous real functions $\mathcal{G}_k^*$ on the real line, $1 \leq k \leq K$, such that $\mathcal{G}_k^*\phi$ has a finite integral, $|\mathcal{G}| \leq \otimes_{k=1}^K \mathcal{G}_k^*$. This result ensures that the integrations required in the program for adaptive quadrature can be made arbitrarily accurate by letting all Q_k be sufficiently large.

In IRT, the number Q_k of quadrature points needed for $1 \leq k \leq K$ for adequate integral approximation via adaptive quadrature is normally quite small, especially for relatively high dimension K (Schilling & Bock, 2005). The basis of this result involves use of asymptotic expansions related to Laplace approximations (de Bruijn, 1970, chapter 4). Results are most favorable if $G_j = H_j$ for each item j, and it is quite helpful in typical cases if the number J_i of items presented to an examinee i is large relative to the dimension K. It is common for values of Q_k from 3 to 5 to be adequate for practical work, and $Q_k = 2$ may be satisfactory for tests for which J_i is large.

3.7 Normal Weights and Even Spacing

A simple alternative approach to Gauss–Hermite quadrature also relies on integers $Q_k > 1$ defined for $1 \le k \le K$. Corresponding to each integer $Q > 1$ is a set $\mathcal{E}_Q$ with Q real members such that, for some real $e_Q > 0$, $\mathcal{E}_Q$ consists of the real numbers $e_Q[q - (Q + 1])2]$ for positive integers $q \le Q$. The function $\mathcal{V}_Q$ is then defined so that

$$\mathcal{V}_Q(\mathcal{U}) = \frac{\phi(\mathcal{U})}{\sum_{\mathcal{U}' \in \mathcal{E}_k} \phi(\mathcal{U})}. \tag{94}$$

The set $\mathcal{Q}$ for adaptive quadrature is the $\prod_{k=1}^{K} \mathcal{E}_{Q_k}$. The weight function $\mathcal{W}$ satisfies

$$\mathcal{W} = \prod_{k=1}^{K} \mathcal{V}_{Q_k}. \tag{95}$$

Weak convergence results again can be applied to $\mathcal{I}_{Q\mathcal{W}}$ (Rao, 1973, chapter 2). Let e_Q approach 0 and $e_Q Q$ approach ∞ as Q approaches ∞. If each Q_k approaches ∞, then rather similar arguments to those used for Gauss–Hermite quadrature can be used to demonstrate that $\mathcal{I}_{WQ}(\mathcal{G})$ converges to the Lebesgue integral of $\mathcal{G}\phi_{KV}$ when $\mathcal{G}$ is $\mathcal{G}\phi_{KV}$ has a finite integral, $\mathcal{G}$ is continuous almost everywhere, and $\mathcal{G}/\phi_{KV}^d$ is bounded for some positive integer $d < 1$.

The program permits e_Q to be specified by the user or to be determined automatically. In the case of automatic specification, e_Q is $2/(Q-1)^{1/3}$. If each Q_k is 2 and e_Q is selected automatically, then quadrature with normal weights and even spacing is the same as Gauss–Hermite quadrature. The choice of e_Q is based on an examination of $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{H}_2)$ and $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{H}_4)$ for this choice for a selection of values of Q. Recall that $\int \mathcal{H}_q \phi = 0$ for $q \ge 1$, and note that symmetry implies that $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{H}_q)$ is 0 for q a positive odd integer. Thus $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{H}_2)$ and $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{H}_4)$ should be near 0 if $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{G})$ is to provide good approximations for $\int \mathcal{G}\phi$ for real functions $\mathcal{G}$ such that $\mathcal{G}\phi$ is integrable and $\mathcal{G}$ has at least five continuous derivatives. To illustrate the quality of the approximation in the case of $Q = 10$, observe that $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{H}_2)$ is about -0.00002 and $\mathcal{I}_{\mathcal{V}_Q \mathcal{E}_Q}(\mathcal{H}_4)$ is about -0.0004.

In practice, the option for even spacing is most likely to be employed with a polytomous latent vector (Haberman et al., 2008).

3.8 Simplex Quadrature

Even with values of Q_k of 3 or 4, as the dimension K increases, the set $\mathcal{Q}$ for Gauss–Hermite quadrature or quadrature with normal weights and even spacing becomes quite large, for $\mathcal{Q}$ has

$\prod_{k=1}^{K} Q_k$ members. As a consequence, it is appropriate to seek quadrature approaches that use very few points. Simplex quadrature provides a very simple illustration of a possible approach that only requires $K+1$ quadrature points. Let $\mathbf{c}_K$, c real, be the K-dimensional vector with all elements c. Let $\mathbf{I}_K$ be the K by K identity matrix. Then let $\mathcal{Q}$ be a set of $K+1$ vectors of dimension K such that the following orthonormality conditions hold:

$$\sum_{\mathcal{U}\in\mathcal{Q}} \mathcal{U} = \mathbf{0}_K \tag{96}$$

and

$$\sum_{\mathcal{U}\in\mathcal{Q}} \mathcal{U}\mathcal{U}' = \mathbf{I}_K. \tag{97}$$

For $K=1$ and $Q_1=2$, simplex quadrature is the same as Gauss–Hermite quadrature. For $K>1$, the set $\mathcal{Q}$ must be selected. The normal selection in the program is based on Helmert contrasts. The set $\mathcal{Q}$ includes $\mathcal{U}_k$, $1 \le k \le K+1$, where element j of $\mathcal{U}_k$, $1 \le j \le K$, is

$$\mathcal{U}_{jk} = \begin{cases} \{(K+1)/[j(j+1)]\}^{1/2}, & j \ge k, \\ -[(K+1)j/k]^{1/2}, & j = k-1, \\ 0, & j < k-1. \end{cases} \tag{98}$$

3.9 Cross-Polytope Quadrature

To define cross-polytope quadrature, let $\boldsymbol{\delta}_k$, $1 \le k \le K$, be the K-dimensional vector with element k equal to 1 and other elements 0. Let $\mathcal{Q}$ be the set that consists of the $2K$ vectors $K^{1/2}\boldsymbol{\delta}_k$ and $-K^{1/2}\boldsymbol{\delta}_k$ for $1 \le k \le K$. Then $\mathcal{Q}$ is the set of vertices of a cross-polytope (Coxeter, 1963). With $\mathcal{W}$ equal to the function on $\mathcal{Q}$ always equal to $(2K)^{-1}$, one finds that $\mathcal{S}$ in (85) includes all polynomials on R^K of degree no greater than 3. If $K=1$, then cross-polytope quadrature is the same as simplex quadrature.

3.10 Variants of Gauss–Hermite Quadrature

In a variety of cases, a reduced set of quadrature points is used based on Gauss–Hermite quadrature. For example, one might consider the case of $Q_k = 3$ for $1 \le k \le K = 3$. One may consider a set $\mathcal{Q}$ with 19 rather than 27 points. The elements of $\mathcal{Q}$ include the zero vector $\mathbf{0}_3$, the 6 vectors $3^{1/2}c\boldsymbol{\delta}_k$, c equal -1 or 1, $1 \le k \le 3$, and the 12 vectors $3^{1/2}(c_k\boldsymbol{\delta}_k + c_m\boldsymbol{\delta}_m)$, c_k and c_m

each either -1 or 1, $1 \leq k < m \leq 3$. The function $\mathcal{W}$ assigns weight 1/3 to $\mathbf{0}_3$, weight 1/18 to $3^{1/2}c\boldsymbol{\delta}_k$, and weight 1/36 to the remaining 12 vectors in $\mathcal{Q}$. Note that in section 3.5, for $\mathcal{T}_k = \mathcal{N}_k$ and $\mathcal{Y}_k = \mathcal{V}_k$, one may define $\mathcal{A}(\mathcal{U})$ so that the function $\mathcal{W}$ assigns weight 1/3 to $\mathbf{0}_3$, weight 1/18 to $3^{1/2}c\boldsymbol{\delta}_k$, and weight 1/36 to the remaining 12 vectors in $\mathcal{Q}$. In this manner, $\mathcal{S}$ in (85) includes all polynomials on R^3 of degree no greater than 5.

An alternative approach can be based on fractional factorial designs for experiments (Box, Hunter, & Hunter, 2005). One can select a set $\mathcal{Q}$ of K-dimensional vectors with all elements either -1 or 1. For some positive integer $K' < K$, $\mathcal{Q}$ has $2^{K-K'}$ elements, and $\mathcal{W}$ is the constant function on $\mathcal{Q}$ with value $2^{K'-K}$. For example, one might have $K = 8$, $K' = 1$, and $\mathcal{Q}$ equal to the set of K-dimensional vectors $\mathcal{U}$ with elements $\mathcal{U}_k$ such that $|\mathcal{U}_k| = 1$ for $1 \leq k \leq K$ and such that $\sum_{k=1}^{K} \mathcal{U}_k$ is even. In (85), $\mathcal{S}$ includes all polynomials on R^8 of degree no greater than 3.

4 Input and Output Specifications

Input for the program consists of a data file which contains the observations and a control file which follows Fortran 2003 rules for namelist input. The program is run from a command line. Within a Unix/Linux environment (including MacOS), the command line is opened by opening a terminal. In Windows, a command prompt is used to obtain a command line. The user is expected to be able to change directories and perform other basic tasks associated with the basic commands of an operating system. Owing to weaknesses in memory management in Windows, program performance is typically much better in a Linux/Unix environment.

The program name is mirt, and piping is normally employed on the command line to specify a control file. If control.txt is the name of the control file and if the executable file is in the path used to find commands, then the program is invoked with the following command:

```
mirt<control.txt
```

The control file uses namelist input records to specify the data, the model parameters, the data files, other input files, and output files. Output generally is designed to produce files with comma-separated values (csv files) readily treated by standard software for spreadsheets. A graphical user interface to simplify input and output is currently under preparation.

Each namelist input record contains a name of a namelist group and pairs of variable names and variable values. The record begins with an ampersand and is immediately followed by the name of the namelist group. One or more blank spaces must follow the name of the group. Zero or more pairs of variable names and values follow, and the record is terminated by a forward slash (/). In each pair of variable names and values, the variable name and value are separated by an equals sign. A user not very familiar with Fortran namelist input should avoid use of exclamation points and use of forward slashes not intended to terminate a record. The order of the pairs of names and values does not matter, and a pair need not appear if the default variable value is acceptable. It is prudent to surround a character value by apostrophes, although a pair of quotation marks may also be used. The program terminates if a namelist group is not successfully read. As indicated in the discussion of the various namelist groups, the program may also terminate if namelist variables are invalid.

In this section, the primary data employed for illustrative purposes are a language test in which four skills are tested. For most examples, data from a single test administration are employed. The test includes 34 dichotomous items that measure listening proficiency, 42 items that measure reading proficiency, 6 attempts that measure speaking proficiency, and 2 items that measure writing proficiency. The listening items all have scores 0 or 1. Of the 42 reading items, 39 are dichotomous items with scores 0 or 2 and 3 are trichotomous items with possible scores 0, 1, or 2. The original speaking items have 5 possible scores from 0 to 4; however, the scores are transformed prior to analysis by subtracting 1 from each positive score. The writing items have original scores of 0 and the integers 2 to 10; however, 1 is subtracted from each score prior to analysis. Generally the 34 listening items are used to illustrate approaches for a test with a one-dimensional latent variable and dichotomous responses. The speaking items are used in some instances to illustrate approaches for a test with a one-dimensional latent variable and with polytomous responses. The complete test illustrates use of methods for a test with both dichotomous and polytomous items and with an underlying latent vector of dimension greater than 1. In many cases, the complete test is used with an underlying latent vector of dimension 4. The data are found in FORM1ANOV.TXT. Data from a different administration of the same test are employed to illustrate analysis when two groups of examinees are involved. These data are in form3cg.txt. These data have a slightly different format, for the writing section scores have already been transformed to the scale from 0 to 9, and there are 41 reading items, of which two trichotomous items have scores from 0 to 2 and one item has four scores from 0 to 3. This section example is used with models with a latent vector with two dimensions, one of which corresponds to all skills except speaking and the other of which corresponds to speaking. The control file listening.txt and associated output file listening.csv provide an example of a simple 2PL analysis with a normal latent variable with mean 0 and variance 1. The control file fourskill.txt with associated output file fourskill.csv illustrates a simple 2PL analysis with a normal latent vector for a between-item model.

The following namelist group names are used. They are arranged in the order they appear in the control file.

- runtitle
- numberunits
- files
- units
- dataspec
- numberrecodes
- recodetable
- paramspec
- dimension
- linearspec
- quadsize
- nquadperdim
- griddata
- quaddim
- quadrature
- allfactorspecs
- factorspecs

- allskillspecs
- skillspecs
- allitemspecs
- itemspecs
- catspecs
- intspecs
- slopespecs
- predictorname
- designparameters
- designspecs
- constraints
- readgamma
- inputinformation
- printprogress
- output
- eapoutput
- weightedsum
- numberweights
- readweight

Variables in each namelist group are defined in the subsection for that group.

4.1 runtitle

This namelist group has the following variable:

- title

4.1.1 title

This variable is a character variable with length up to 80 characters. The default value, 'GPCM_2PL_Model_ with_ Normal_Latent_Distribution', corresponds to the default analysis. In listening.txt, 'Listening_Test:2PL_Model' is the value of title in the runtitle group. Were no entry specified for title, then the default title would be used.

4.1.2 Remarks.

Because output is for files with comma-separated values (csv files), it is prudent to avoid commas, spaces, apostrophes, and quotation marks within the run title. This issue applies to all names used in the program; however, no error is reported if spaces are included.

4.2 numberunits

This namelist group has the following variable:

- count

4.2.1 count

This variable is an integer from 1 to 90. The default value is 1. It specifies the number of files employed for program output. In listening.txt, the default value is used, for no value is specified for count. In contrast, in listening1.txt, count is 9, for 9 separate output files are specified.

4.3 files

There are count namelist groups, one for each output file. The ith group corresponds to Fortran unit $9+i$. Corresponding to the namelist group is the following variable:

- filename

4.3.1 filename

The value of filename is the name of a file. To permit use of paths, this value is a character variable with up to 256 characters. The default value of filename is ‘output.csv’. If the file specified cannot be opened, then the program terminates with an error message. Note that one file should not be assigned to multiple units. Examples of filename specifications can be seen in listening.txt, where all output is placed in ‘listening.csv’, and listening1.txt, where nine separate output files are specified.

4.4 units

This namelist group has the following variables to specify Fortran units associated with specific portions of the output. For each variable, the default value is 10, the unit which corresponds to the name of the first output file. In listening.txt, this default value is used for all output; however, output is somewhat more complex in listening1.txt. The portions of the output correspond to the following variables:

- unitalpha
- uniteap
- uniteapskill
- uniteapwt
- unitgrad
- unitinfo
- unititeration
- unititerationstart
- unitmargin
- unitmarginwtsum
- unitmargin2

- unitmp
- unitoutdata
- unitparam
- unitparamcov
- unitparamcov_complex
- unitparamcov_louis
- unitparamcov_sandwich
- unitpost
- unitpreditem
- unitprob
- unitrel
- unitrelskill
- unitrelwt
- unittitle
- unitwtitem

4.4.1 unitalpha

The unit for $\boldsymbol{\theta}_{im}$ and $\mathbf{L}_i$. The specification for $\mathbf{L}_i$ involves a K by K matrix $\bar{\mathbf{L}}_i$ such that the element in row k and column $k' < k$ of the matrix is the element in row k and column k' of $\mathbf{L}_i$ divided by the element in row k' and column k' of $\mathbf{L}_i$, and the element in row $k' \leq k$ and column k of the matrix is the element in row k and column k' of $\mathbf{L}_i$ multiplied by the element in row k' and column k' of $\mathbf{L}_i$. This unit is only useful for adaptive quadrature. It can assist in recalculations. In listening1.txt, unitalpha is not specified and takes its default value of 10. In listeninga.txt, the unit is 11, which corresponds to listeningalpha.csv.

4.4.2 uniteap

The unit for the estimated conditional expected (expected a priori [EAP]) value $\hat{\boldsymbol{\theta}}_i$ of $\boldsymbol{\theta}_i$ given X_{ij}, j in $\mathcal{J}_i \cap \mathcal{J}'$ and $\mathbf{Z}_i$ and the estimated covariance matrix of $\boldsymbol{\theta}_i$ given X_{ij}, j in $\mathcal{J}_i \cap \mathcal{J}'$, and $\mathbf{Z}_i$ for each examinee i, where $\mathcal{J}'$ is a subset of $\mathcal{J}$ that may be equal to $\mathcal{J}$ (Bock & Aitkin, 1981; Haberman & Sinharay, 2010). In listening1.txt, uniteap is 11. The corresponding file is listeningeap.csv. Because the latent variable has dimension 1 in this example, and because observations do not have identifiers such as customer numbers, the three columns consist of a sequence number, an EAP ("Mean_Listening") and an estimated conditional variance ("Cov_Listening_Listening") of the latent variable given the item responses. Note that the conditional covariance of Listening and Listening shown in the output is obviously the same as the conditional variance of Listening.

4.4.3 uniteapskill

The unit for the estimated conditional expected value of $\mathbf{A}\boldsymbol{\theta}_i$ given X_{ij}, j in $\mathcal{J}_i \cap \mathcal{J}'$ and $\mathbf{Z}_i$, and the estimated covariance matrix of $\mathbf{A}\boldsymbol{\theta}_i$ given X_{ij}, j in $\mathcal{J}_i \cap \mathcal{J}'$, and $\mathbf{Z}_i$ for each examinee i. In listening1.txt, uniteapskill is not specified and takes its default value of 10. In this case, no request is ever made for the transformed EAP values, for the dimension of the $\boldsymbol{\theta}_i$ is 1. In threefacte.txt, transformed EAP values are requested. Here uniteapskill is 11. The corresponding file is threefacteapskill.csv. In this example, the skills are Listening, Reading, Speaking, and Writing. To each observation corresponds a sequence number, 4 conditional means, and 16 conditional covariances.

4.4.4 uniteapwt

The unit for the estimated conditional expected value $\widehat{\mathbf{TS}}_i$ of $\mathbf{TS}_i$ and conditional covariance matrix of $\mathbf{TS}_i$ given $\mathbf{X}_i$ and $\mathbf{Z}_i$, $1 \le i \le n$, where $\mathbf{TS}_i$ is the sum $\sum_{j \in \mathcal{J}'} \mathbf{IS}_j(Y'_{ij})$, $\mathbf{IS}_j$, j in $\mathcal{J}'$, is a specified DS-dimensional vector function on the integers 1 to H_j, and the Y'_{ij}, j in $\mathcal{J}'$, are random variables such that $1 \le Y'_{ij} \le H_j$ and such that the Y'_{ij}, j in $\mathcal{J}'$, and $\mathbf{Y}_i$ are conditionally independent given $\boldsymbol{\theta}_i$ and $\mathbf{Z}_i$. The conditional expected value of $\mathbf{TS}_i$ given $\boldsymbol{\theta}_i$ and $\mathbf{Z}_i$ is denoted by $\widetilde{\mathbf{TS}}_i$, and $\widehat{\mathbf{TS}}_i$ is the conditional expected value of $\widehat{\mathbf{TS}}_i$ given $\mathbf{X}_i$ and $\mathbf{Z}_i$. As in (9), it is assumed that the conditional probability that Y'_{ij} equals y, $1 \le y \le H_j$, given $\boldsymbol{\theta}_i = \boldsymbol{\omega}$ in Ω and $\mathbf{Z}_i = \mathbf{z}$ in $\mathcal{Z}$ is $p_{Yj}(y|\boldsymbol{\omega})$ (Haberman & Sinharay, 2010). In listening1.txt, uniteapwt is 19. The corresponding file is listeningeapwt.csv. The sum of the item scores for Listening is considered. The format is the same as in listeningeap.csv; however, the scale of results is quite different, for the sums are integers from 0 to 34.

4.4.5 unitgrad

The unit for the gradients $\nabla\ell_i(\hat{\gamma})$ from the observations i. In listening1.txt, unitgrad is 14. The corresponding file is listeninggrads.csv. To each observation corresponds a sequence number and the 68 elements of the gradient. Column labels identify the parameter corresponding to the column, so that "Listening_item27_slope" corresponds to the slope parameter for Item 27. This output is generally only relevant for methods of analysis not incorporated into the program.

4.4.6 unitinfo

The unit used for a summary of information measures used to assess model performance. The model dimension is C if $\nabla^2\ell_V(\hat{\gamma})$ is negative definite. Otherwise, the model dimension is the rank of $\nabla^2\Phi_V(\hat{\gamma})$. The estimated expected log penalty per presented item

$$\mathrm{PE} = \frac{-\ell(\hat{\gamma})}{\sum_{i=1}^{n} w_i J_i} \tag{99}$$

is supplied. The measure is between 0 and the logarithm of the largest number of observed categories associated with an item, so that the measure cannot exceed log(2) if all items are dichotomous. This measure has been applied to IRT in the case of equal weights and $\mathcal{J}_i$ constant (Sinharay, Haberman, & Lee, 2011). Without the normalization by number of items, the measure has been employed for model evaluation outside of IRT (Gilula & Haberman, 1994, 1995, 2001). The logarithmic penalty function itself has a much longer history in statistics (Mosteller & Wallace, 1964; Savage, 1971). The estimated asymptotic standard error of PE is the same as in the case of chained log-linear models for multinomial responses (Gilula & Haberman, 1995). In addition, a version AK of PE based on the Akaike (1974) approach is provided in which the model dimension is added to the numerator in (99), and a version GH of PE based on the Gilula–Haberman approach (Gilula & Haberman, 1994, 1995) is provided in which the trace of $[-\nabla^2\ell_V(\hat{\gamma})]^{-1}\mathbf{\Phi}(\hat{\gamma})$ is added to the numerator in (99) if $-\nabla^2\ell_V(\hat{\gamma})$ is nonsingular. If the Louis approximation to the negative Hessian matrix is used in computation of maximum likelihood estimates, the GH and AK are the same. If complex sampling is used, then the Gilula–Haberman version of PE is also calculated with $\mathbf{\Phi}(\hat{\gamma})$ replaced by the estimated covariance matrix $\widehat{\mathrm{Cov}}(\nabla\ell(\boldsymbol{\gamma}))$ of $\nabla\ell(\boldsymbol{\gamma})$ based on complex sampling. Complex sampling options are described in section 4.5. In listening1.txt, unitinfo is not specified and takes its default value of 10. For sample output, see rows 23 to 28 of listening.csv. The penalty label corresponds to the estimated expected log penalty per presented item. The label "SE_Penalty" corresponds to the standard error of this estimate. This standard error applies to the Gilula–Haberman and Akaike measures as well. Owing to the large sample size, all penalty estimates are quite similar and close to 0.5, a value somewhat lower than the upper bound of $\log(2) = 0.693$. The Gilula–Haberman and Akaike measures always exceed the basic penalty estimate. As in the example, the Gilula–Haberman and Akaike measures are usually quite close, especially in large samples. They sometimes differ more noticeably when models fit badly.

4.4.7 unititeration

The unit for summary information concerning iterations using the full sample. For each iteration t, the step size α_t and the log likelihood $\ell(\boldsymbol{\gamma}_{t+1})$ are recorded. As will be discussed under the namelist record for the group printprogress, this unit is not relevant if iteration information is sent to standard output. In listening.txt, unititeration is not specified and takes its default value of 10, so that results are in listening.csv. In listening.csv, the summary is found in rows 18 to 22. There are three iterations, each step size is 1, and the log-likelihood changes from -162545 to -162543.

4.4.8 unititerationstart

The unit for summary information concerning initial iterations using a systematic sample of observations. The output is the same as for unititeration, save for the initial label for the iteration data. In listening.txt, unititerationstart is not specified and takes its default value of 10, so that results are in listening.csv. In listening.csv, the summary is found in rows 9 to 17. There are seven iterations, each step size is 1, and the log-likelihood changes from -17897.1 to -16902.3. Note that the log-likelihood for initial iterations is much smaller than for the regular iterations (rows 18 to 22) due to use of a systematic sample of size 1,000 rather than the full sample of 9,617 examinees.

4.4.9 unitmargin

The unit for summary information concerning marginal distributions of individual item responses. If requested, output includes observed and fitted frequency distributions, reliability coefficients of individual item responses. If printmarginres is .TRUE. in output, then adjusted residuals (or generalized residuals) are also provided. Adjusted residuals are computed throughout the program in a manner quite similar to the approach in Haberman (2009) and Haberman and Sinharay (in press). They are constructed to have approximate standard normal distributions in large samples if the model is satisfied. In listening1.txt, unitmargin is 16, so that results are in listeningmarg.csv. The columns specify item names, category numbers, item frequencies, observed category frequencies, standard errors for observed category frequencies, fitted category frequencies, observed category proportions (or relative frequencies), standard errors of observed category proportions, fitted category proportions, standard deviations of category indicators, standard errors of measurement of category indicators, reliabilities of category indicators, residuals for category frequencies, standard errors for these residuals, residuals for category proportions, standard errors for these residuals, and the common adjusted residuals for both category frequencies and category proportions. For example, for Category 1 of Item 1, the item was presented to all 9,617 examinees and was answered correctly by 6,876 of them. The standard error of the number answered correctly was about 44. As should be the case for any 2PL model with no restrictions on item parameters, the fitted category frequency is essentially the same as the observed category frequency. Any discrepancy simply reflects the fact that the algorithm terminates after a finite number of iterations, so that $\hat{\gamma}$ is not computed exactly. The observed category proportion is 0.7150, so that 71.5% of examinees provided a correct response to this item. The standard error associated with this proportion is 0.0046, so that the proportion is relatively well determined. The standard deviation 0.451 of the category indicator is the estimated standard deviation of the indicator variable, which is 1 for a correct answer and 0 otherwise. The standard error of measurement 0.428 for this category indicator ("Std_err_cat_ind") is the square root of the estimated conditional variance of the indicator variable given the latent variable. The reliability 0.102 of the category indicator is $1 - (0.428/0.451)^2$. It is the estimated reliability of the category indicator when regarded as a test with a single item. The residual for the category frequency, the standard error for the residual, the residual for the category proportion, and the standard error for this residual are all close to 0, for the observed and fitted quantities are the same, except for errors in numerical approximation. As a consequence, the adjusted residual is reported to be 0. For a case in which the adjusted residuals are more notable, consider writing.txt and writing.csv. Here a test of writing with two items is considered with 10 categories per item. A quadratic model is used for logarithms of conditional probabilities of item categories given the latent variable, so that the estimated expected frequencies in writingmarg.csv may not equal the corresponding observed values. Indeed, there are clearly substantial discrepancies between observed values and estimated expected values, and quite large adjusted residuals are found in a number of cases.

4.4.10 unitmarginwtsum

The unit for a summary for observed and fitted frequency distributions for weighted sums of item scores. Output includes observed and fitted frequency distributions for the marginal weighted sums specified in weightedsum. If printmarginwtsum is .TRUE. in output, then adjusted residuals are also provided. In listening1.txt, unitmarginwtsum is 18, so that output is found in listeningmargwtsum.csv. This output is for the sum of Listening scores for the 34 Listening items. Summaries are provided for exact weighted sums and for cumulative weighted sums. For example, consider the entry for the score 10. There are 9,617 examinees for which the score could be computed. Of these examinees, 125 have a sum score of 10, and the standard error associated with this frequency is 11.1. On the other had, 376 examinees had a sum score of 10 or less, and the corresponding standard error of this frequency was 19.0. The estimated expected number of examinees with a score of 10 is 112.3, while the estimated expected number of examinees with a score of 10 or less is 403.5. In terms of proportions or relative frequencies, the observed fraction of examinees with a score of 10 is 0.0130, and the corresponding standard error is 0.0036. The observed fraction of examinees with a score of 10 or less is 0.0391, and the corresponding standard error is 0.0020. The estimated probability of a score of 10 is 0.0117, and the estimated probability of a score of 10 or less is 0.0420. The residual for the number of examinees with a score of 10 is $12.7 = 125 - 112.3$, and the associated standard error is 10.2. The residual for the number of examinees with a score of 10 or less is -27.5, and the corresponding standard error is 10.3. In terms of proportions, the residual for the fraction of examinees with a score of 10 is 0.00132, and the corresponding standard error is 0.00102. The residual for the fraction of examinees with a score of 10 or less is -0.00286, and the corresponding standard error is 0.00107. The resulting adjusted residuals are 1.25 for the number (or fraction) of examinees with a score of 10 and -2.67

for the number (or fraction) of examinees with a score of 10 or less. The latter adjusted residual suggests some problem with model error, although a comparison of observed and fitted values suggests that the actual discrepancy is not very large.

4.4.11 unitmargin2

The unit for a summary for observed and fitted frequency distributions of pairs of item responses. If printmargin2res is .TRUE., then adjusted residuals are also provided. In listening1.txt, unitmargin2 is 17, and the corresponding file is listeningmarg2.csv. The table format is rather similar to the format for marginal frequencies described in unitmargin. For example, consider the entry for Item 1, category 1, and Item 2, category 1. Here 9,617 examinees were presented with both items, and 5,942 answered both correctly. The corresponding standard error is 47.7. The estimated expected number of examinees with this combination of responses is 5,850. The observed fraction of examinees with a correct response to both items is 0.6179, and the corresponding standard error is 0.0050. The estimated probability that an examinees answers both items correctly is 0.6083. For the frequency of the combination of responses, the residual is 92.0, and the corresponding standard error is 16.1. For the proportions of examinees with both responses correct, the residual is 0.0096, and the corresponding standard error is 0.0017. The adjusted residual of 5.71 clearly indicates a problem with the model, although the actual size of the residuals does not indicate a very large discrepancy.

4.4.12 unitmp

The unit for output of $\boldsymbol{\theta}_{im}$ and the inverse of $-\nabla^2\mathcal{L}_i(\boldsymbol{\theta}_{im})$. This output is used for maximum a posteriori (MAP) likelihood estimation of the latent vectors $\boldsymbol{\theta}_i$ (Bock & Aitkin, 1981; Lord, 1980). In listening1.txt, unitmp is 15. The output file is listeningmap.csv. The output is similar to listeningeap.csv, although in listeningmap.csv, the second column is the MAP estimate and the third column is the inverse of the negative Hessian matrix at the MAP estimate for the observation specified in the first column.

4.4.13 unitoutdata

The unit for a basic data summary that shows the number n of observations used from the input file, the number J of items in the analysis, the number U of predictors, and the filename of the input file. In listening.txt, unitoutdata is not specified and takes its default value of 10. For sample output, see rows 1 to 8 in listening.csv. Here one finds that 9,617 observations are available, 34 items are present, there is only one predictor, the constant predictor, the data file is 'FORM1ANOX.TXT', the total numbers of items presented to examinees is $9617 \times 34 = 326978$, and the average number of items presented per examinee is 34, for each examinee is presented with each item.

4.4.14 unitparam

The unit used for a list which, for each integer c from 1 to C includes the name of parameter γ_c, the estimated parameter $\hat{\gamma}_c$, the estimated asymptotic standard deviation $\hat{\sigma}(\hat{\gamma}_c)$ of $\hat{\gamma}_c$ derived from the square root of row c and column c of the estimated asymptotic covariance matrix

$$\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\gamma}}) = [-\nabla^2 \ell_V(\hat{\boldsymbol{\gamma}})]^{-1}[-\nabla^2 \ell(\hat{\boldsymbol{\gamma}})][-\nabla^2 \ell_V(\hat{\boldsymbol{\gamma}})]^{-1} \tag{100}$$

of $\hat{\boldsymbol{\gamma}}$, the estimated asymptotic standard deviation $\hat{\sigma}_L(\hat{\gamma}_c)$ of $\hat{\gamma}_c$ derived from the square root of row c and column c of the Louis (1982) estimate

$$\widehat{\mathrm{Cov}}_L(\hat{\boldsymbol{\gamma}}) = [\boldsymbol{\Phi}_V(\hat{\boldsymbol{\gamma}})]^{-1}\boldsymbol{\Phi}(\hat{\boldsymbol{\gamma}})[\boldsymbol{\Phi}_V(\hat{\boldsymbol{\gamma}})]^{-1} \tag{101}$$

of the asymptotic covariance matrix of $\hat{\boldsymbol{\gamma}}$, and the estimated asymptotic standard deviation $\hat{\sigma}_S(\hat{\gamma}_c)$ of $\hat{\gamma}_c$ derived from the square root of row c and column c of the sandwich estimated asymptotic covariance matrix

$$\widehat{\mathrm{Cov}}_S(\hat{\boldsymbol{\gamma}}) = [-\nabla^2 \ell_V(\hat{\boldsymbol{\gamma}})]^{-1}\boldsymbol{\Phi}(\hat{\boldsymbol{\gamma}})[-\nabla^2 \ell_V(\hat{\boldsymbol{\gamma}})]^{-1} \tag{102}$$

obtained without the assumption that the model holds (Haberman, 1989; Huber, 1967; White, 1980). If the Louis approximation is used for the negative Hessian matrix in implementation of the Newton–Raphson algorithm, then these three estimated asymptotic standard deviations are all the same. If complex sampling is used, then the estimated asymptotic standard deviation $\hat{\sigma}_C(\hat{\gamma}_c)$ of $\hat{\gamma}_c$ is derived from the square root of row c and column c of the estimated asymptotic covariance matrix

$$\widehat{\mathrm{Cov}}_S(\hat{\boldsymbol{\gamma}}) = [-\nabla^2 \ell_V(\hat{\boldsymbol{\gamma}})]^{-1}\widehat{\mathrm{Cov}}(\nabla \ell(\boldsymbol{\gamma}))[-\nabla^2 \ell_V(\hat{\boldsymbol{\gamma}})]^{-1}. \tag{103}$$

In listening.txt, unitparam is not specified and takes its default value of 10. An example of output appears in rows 29 to 98 of listening.csv. In this example, all expressions for standard errors yield rather similar results. The third column is $\hat{\sigma}(\hat{\gamma}_c)$, the fourth column is $\hat{\sigma}_L(\hat{\gamma}_c)$, and the fifth column is $\hat{\sigma}_S(\hat{\gamma}_c)$. In listeningp.txt, an artificial example of complex sampling is provided in which primary sampling units are defined in terms of a sequence number. In this case, unitparam is not specified, so that unit 10 is used for output. The corresponding file is listeningp.csv. Rows 31 to 100 contain the information on parameter estimates. Because the example is artificial, the added column for $\hat{\sigma}_C(\hat{\gamma}_c)$ is quite similar to the other columns.

4.4.15 unitparamcov

The unit for the estimated asymptotic covariance matrix $\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\gamma}})$. In listening1.txt, unitparamcov is 12, and the corresponding file is listeningcov.csv.

4.4.16 unitparamcov_complex

The unit for the estimated asymptotic covariance matrix $\widehat{\mathrm{Cov}}_C(\hat{\boldsymbol{\gamma}})$. In listeningp1.txt, unitparamcov_complex is 11, and the corresponding file is listeningparamcovp.csv.

4.4.17 unitparamcov_louis

The unit for the estimated asymptotic covariance matrix $\widehat{\mathrm{Cov}}_L(\hat{\boldsymbol{\gamma}})$. In listeningcl.txt, unitparamcov_louis is 11, and the corresponding file is listeningcovl.csv.

4.4.18 unitparamcov_sandwich

The unit for the estimated asymptotic covariance matrix $\widehat{\mathrm{Cov}}_S(\hat{\boldsymbol{\gamma}})$. In listeningcs.txt, unitparamcov_sandwich is 11, and the corresponding file is listeningcovs.csv.

4.4.19 unitpost

The unit for the estimates, for each examinee i, of the posterior distribution. For the case of latent classes, the posterior probability is estimated that $\boldsymbol{\theta}_i = \boldsymbol{\omega}$ in Ω given the observed responses $\mathbf{X}_i$. In the normal case, the posterior probability is estimated that $\boldsymbol{\theta}_i$ is equal to $\mathcal{U}$ in $\mathcal{Q}_i$ given that $\boldsymbol{\theta}_i$ is in $\mathcal{Q}_i$ and given the observed responses $\mathbf{X}_i$. In listening1.txt, unitpost is 13, and the corresponding file is listeningpost.csv. Output includes the observation identification and pairs of locations and weights. Thus for the first examinee, the first location is -0.286 and the first weight is 0.00313.

4.4.20 unitpreditem

The unit for totals and averages of products of category indicators and predictors. The default value is 10. In Four3C25twog.txt, the value 12 is used, and the corresponding file is Four3Cn25twogpreditem.csv. In this example, the predictor is an indicator for membership in Group 2, where examinees are in Group 1 or Group 2. Thus the row for category 1 of Item 1 indicates that the item was presented to 21,238 examinees, and 7,897 of these examinees answered the item correctly and also belonged to Group 2. The standard error of this number of correct responses from Group 2 is 70. Under the model used in the example, the estimated expected number of examinees who responded correctly and were in Group 2 is 8,537. A similar analysis is expressed in terms of fractions. The fraction of examinees presented the item who answered correctly and were in Group 2 was 0.3718, and the corresponding standard error is 0.0004. The corresponding estimated expected fraction under the model is 0.4020. The residual for the number of examinees with the correct answer and membership in Group 2 is −640, and the corresponding standard error is 23. For proportions, the residual is −0.0302, and the corresponding standard error is 0.0011, so that the adjusted residual is −27.3, a very small value for an adjusted residual. It follows clearly from this adjusted residual and others in the file that failure to account for group membership induces an appreciable model error.

4.4.21 unitprob

The unit for the estimate for each examinee i of the probability $p(\mathbf{X}_i)$. In listening3.txt, unitprob is 11, which corresponds to listeningprob.csv. Output consists of the observation identification and the probability. This output can be used outside the program for various model comparisons of the type described in Gilula and Haberman (2001).

4.4.22 unitrel

The unit for the reliability coefficients for the elements $\hat{\theta}_{ik}$ of $\hat{\boldsymbol{\theta}}_i$, $1 \leq k \leq K$ (Haberman & Sinharay, 2010). In listening1.txt, unitrel is not specified and takes its default value of 10, which corresponds to file listening1.csv. The output is found on rows 99 to 110. The average of conditional means is the average of the EAP values $\hat{\boldsymbol{\theta}}_i$ for $1 \leq i \leq n$. The covariance matrix of conditional means is the sample covariance matrix (not corrected for bias) of the $\hat{\boldsymbol{\beta}}_i$. The conditional covariance matrix is the average of the estimated conditional covariance matrices of $\boldsymbol{\theta}_i$

given $\mathbf{X}_i$ and $\mathbf{Z}_i$ for $1 \leq i \leq n$. The estimated unconditional covariance matrix for $\boldsymbol{\theta}_i$ is the sum of the conditional covariance matrix of conditional means and the conditional covariance matrix. For $1 \leq k \leq K$, the reliability coefficient of element $\hat{\theta}_{ik}$ of $\hat{\boldsymbol{\theta}}_i$ is then the ratio of the estimated variance of $\hat{\theta}_{ik}$ to the estimated unconditional variance of element θ_{ik} of $\boldsymbol{\theta}_i$. In the example, $K = 1$. As should be the case given a model in which the θ_{i1} are standard normal variables, the average EAP of -0.00174 is quite close to 0, and the estimated conditional variance 0.9987 of θ_{i1} is close to 1. The reliability estimate for the EAP is 0.867, the ratio of 0.866, the estimated variance of $\hat{\theta}_{i1}$, to 0.9987.

4.4.23 unitrelskill

The unit for the reliability coefficients for the DA elements of $\mathbf{A}\hat{\theta}_i$. In threefacte.txt, unitrelskill is not specified and takes its default value of 10. The corresponding output is in threefacte.csv in rows 225 to 253. The format is quite similar to the format described in unitrel; however, results are for the transformed EAP values $\mathbf{A}\hat{\boldsymbol{\theta}}_i$. In threefacte.csv, the skills are Listening, Reading, Speaking, and Writing. The reliability estimates range from 0.917 for Reading to 0.871 for Writing. The estimated variances of the elements of $\mathbf{A}\boldsymbol{\theta}_i$ exceed 1 due to model assumptions.

4.4.24 unitrelwt

The unit for the reliability coefficients for the elements of $\widehat{\mathbf{TS}}_i$. In these reliability coefficients, variances of elements of $\widehat{\mathbf{TS}}_i$ are compared to variances of elements of $\widetilde{\mathbf{TS}}_i$. In listening1.txt, unitrelwt is not specified and takes its default value of 10, and results appear in rows 113 to 126 of listening1.csv. Results are for the sum of the item scores for the 34 Listening items. The format of results is quite similar to the format described in unitrel, except that the estimated means and variances of the sums of the item scores are on a much different scale than are the underlying means and variances of the latent variable.

4.4.25 unittitle

The unit for the title of the analysis. In listening1.txt, unittitle is not specified and takes its default value of 10. The title appears in the first row of listening1.csv.

4.4.26 unitwtitem

The unit for totals and averages of products of category indicators and weighted sums. In listening6.txt, unitwtitem is 11, and the corresponding file is listeningwtitem.csv. The weighted sum is the sum of the 34 item scores. The format is essentially the same as described in unitpreditem. Note that the adjusted residuals suggest model error, although the differences between observed and fitted values are relatively small.

4.5 dataspec

This namelist group has the following variables:

- fileformat
- filename
- nexternal
- nitems
- nobs
- nobsstart
- npred
- complx
- recode
- stratify
- useid
- usepsu
- weight

4.5.1 fileformat

The format for the data file. The default is '*', which corresponds to list-directed input. If the value is not the asterisk '*', then the format is in the standard form for a Fortran 2003 format. The order of reading is examinee identification, item responses, predictors that are not constant, weight, external variables, stratum number, and number of the primary sampling unit within the stratum. Note that not all variables need be present. For example, with the default weighting, the weight variable is the constant 1 and is not read. The examinee identification is a character variable, the item responses are integers, the predictors, weights, and external variables are floating-point numbers, and the numbers for the stratum and primary sampling unit are integers. In listening.txt, 34 item responses are read, and they are located from columns 22 to 55. If the data cannot be read by using fileformat, then the program terminates with an error message. It is important to note that tab coding often permits variables to be read out of order. For example, in listeningp.txt, the sequence 'T4,i2' permits reading of a primary sampling unit (PSU) code in columns 4 and 5 of the input record after the 34 Listening items have been read.

4.5.2 filename

The name of the data file. The default value is 'data.txt'. The name may include path information relative to the directory in which the program was called, so that one may have filename with values such as 'data.txt' or '∼\data\data.txt'. The filename can be up to 256 characters long. The program terminates with an error message if the file cannot be opened. In listening.txt, the filename is 'FORM1ANOX.TXT'.

4.5.3 nexternal

The number of external variables used in construction of summary statistics and residuals. The value should be a nonnegative integer, and the default value of 0 corresponds to no external variables. The value of nexternal is set to 0 if the read value is negative. In listening.txt, nexternal is not specified, so that no external variables are used. On the other hand, nexternal is 1 in Four3Cntwoge.txt. In this example, there are two examinee groups, and a comparison of the observed data and the fitted model examines the association of group membership with indicators for item categories. Results are in Four3Cntwoge.csv, Four3Cn25twogeeap.csv, and Four3Cn25twogepreditem.csv.

4.5.4 nitems

The total number J of items in the analysis. The program terminates with an error message if a positive integer is not supplied. The default setting is 0, so that this entry must be specified. In listening.txt, nitems is 34, so that 34 items are read.

4.5.5 nobs

The number n of observations in the data file. The default is 0. A value of 0 or less results in the number of observations being determined by reading the input data. If nobs is positive and less than the number of observations in the input file, then only the first nobs observations are used in the analysis. The program terminates with an error message if $n \leq 2$. In listening.txt, nobs is not specified, so n is determined by reading the input data. On the other hand, in Four3Cn25twog0.txt, the choice of nobs equal to 10,004 results in an analysis of the data restricted to Group 1. Results are in Four3Cn25twog0.csv. They may be compared with results for complete data in Four3Cntwoge.csv.

4.5.6 nobsstart

The number of observations used for initial determination of starting values. The default value of 1,000 is also used if nobsstart is specified but is not positive. Once the data are read, nobsstart is the minimum of the input value of nobsstart and the number nobs of observations to be analyzed. To simplify computations, the observations are selected by systematic sampling.

4.5.7 npred

The number U of predictors. The value 1 is the default and is also used if npred is less than 1. In listening.txt, npred is not specified, so that $U = 1$. In Four3Cn25twog.txt, two groups are present, so that npred is 2.

4.5.8 complx

A logical variable which is .FALSE. if complex sampling is not used and .TRUE. if complex sampling is used. The default value is .FALSE., and the variable complx is set to .TRUE. if stratify is .TRUE., if usepsu is .TRUE., or if weight is .TRUE. and .TRUE. is also the read value of complx. In Four3Cn25twog1c.txt, .TRUE. is the value of complx; however, this selection has no practical effect, as is evident by examination of Four3Cn25twog1c.csv. Note that the results for standard errors for complex sampling are the same as those for sandwich estimation of standard errors.

4.5.9 recode

A logical variable which is only .TRUE. if some recoding of item responses is required. The value .FALSE. is the default. In speaking.txt, recode is .TRUE., so that recoding is employed.

4.5.10 stratify

A logical variable which is .FALSE., the default, if stratified sampling is not used and .TRUE. if sampling is stratified. In listenings.txt and in listeningps.txt, .TRUE. is the value of stratify. Results appear in listeningps.csv.

4.5.11 useid

A logical variable which is .TRUE. if examinee identifications are used in output that includes results for individual examinees and .FALSE., the default, otherwise. The examinee identification, if present, is a character variable with length no greater than 16. When examinee identification is not provided, a sequence number is used instead in output files for individual examinees. For example, this practice is seen in listeningeap.csv.

4.5.12 usepsu

A logical variable which is .FALSE., the default, if multistage sampling is not used and .TRUE. if multistage sampling is used. In listeningp.txt, .TRUE. is the value of usepsu. This specification affects the values of the estimated asymptotic standard deviations in listeningp.csv reported for complex sampling (Fuller, 2009, pp. 64–68).

4.5.13 weight

A logical variable which is .FALSE., the default, if examinee weights are all 1 and are not to be read. If weight is .TRUE., then the weight variable is read during data input. An error message results if any values of the weight variable are negative. In Four3Cn25twog1.txt, .TRUE. is the value of weight. Because the weight variable is 0 or 1, the practical effect is to confine analysis to Group 2. Results in Four3Cn25twog1.csv may be compared with those in Four3Cn25twoge.csv to illustrate the effects on parameter estimates of confining attention to Group 2. Note that the results for standard errors for complex sampling are the same as those for sandwich estimation of standard errors.

4.6 numberrecodes

If recode is .TRUE. in the namelist group dataspec, then the namelist group numberrecodes is read. This namelist group has the following single variable:

- numbercodes

4.6.1 numbercodes

The variable numbercodes is an integer array with nitems elements. Element j of numbercodes is the number of values of item j which require recoding. Thus this element must be a nonnegative integer. The default value is 0, and any negative value is replaced by 0. If some element of numbercodes is positive, then a namelist input record for the group recodetable is required for each item for which the corresponding element of numbercodes is positive. The records are ordered by increasing item number. In speaking.txt, there are four recodes for each of the six items.

4.7 recodetable

The group recodetable contains the following variable:

- recode_tab

4.7.1 recode_tab

The variable recode_tab is a two-dimensional integer array with two rows and with a number of columns equal to the corresponding element of numbercodes. For each column, the first row is a value of the item that is to be changed to the value in the second row. For example, in speaking.txt, each item is recoded so that code 0 remains 0 but each code from 1 to 4 is reduced by 1.

4.8 paramspec

This namelist group specifies the basic parameters for the numerical algorithm for computation of maximum-likelihood estimates. The following variable names are used:

- maxit
- maxita
- nr
- twostages
- changemin
- kappa
- maxdalpha
- tau
- tol
- tola
- tolres
- tolsing

These specifications are often not modified at all, as is the case in listening.txt.

4.8.1 maxit

The maximum value of t for each stage of iterations. The variable is an integer, and the default value is 50. A choice of maxit of 0 or less results in no iterations. This selection can sometimes be relevant when basic computations have already been performed but some additional output is desired.

4.8.2 maxita

The maximum number of iterations for approximation of the location of the maximum posterior density of $\boldsymbol{\theta}_i$ given $\mathbf{X}_i$ and $\mathbf{Z}_i$. The variable is an integer. If maxita is 0 or less, then quadrature is not adaptive. The default is 10.

4.8.3 nr

The logical variable that indicates whether the stabilized Newton–Raphson algorithm is used. If the indicator has its default value of .TRUE., then the customary stabilized Newton–Raphson algorithm is used. If the indicator is .FALSE., then in (53), $\mathbf{q}_t$ is $[\boldsymbol{\Phi}_V(\boldsymbol{\gamma}_t)]^{-1}\nabla\ell_V(\boldsymbol{\gamma}_t)$ even if $[-\nabla^2\ell_V(\boldsymbol{\gamma}_t)]$ is positive definite. This modification is similar to use of the Fisher (1925) scoring algorithm rather than the Newton–Raphson algorithm in evaluation of maximum-likelihood estimates. The motivation involves reduction in the computations per iteration; however, the number of iterations needed to achieve satisfactory convergence is typically increased. This option is most attractive when the number of items is very large, say, more than 100, or the number of quadrature points is large. The option prevents some analysis of effects of specification error on asymptotic variances and covariances. One comparison can be obtained with fourskill.tex and fourskilll.tex. In the second case, iterations were completed in about one-sixth of the time required in the first case. For comparison of results, see fourskill.csv and fourskilll.csv. Note that the numerical results are slightly different, but the differences have relatively little impact.

4.8.4 twostages

Normally there are two stages to computation of the maximum-likelihood estimates. In the first stage, a subsample of nobsstart observations is used to obtain an approximation to maximum-likelihood estimates for the full sample. In the second stage, computations for the full sample begin with the approximations from the first stage. If twostages is .FALSE., then the first stage is omitted. Otherwise, the first stage is not omitted. In the default case, .TRUE. is the value of twostages. In listeningst.txt, twostages is .FALSE. because the final results of listening.csv are used as input, so that the initial stage serves no purpose.

4.8.5 changemin

In (56), the value of τ_1. The default value of 0.0625 is used if the read value of changemin is not positive or is at least 1/2. The variable is real.

4.8.6 kappa

The maximum permitted value of $\alpha_t|\mathbf{T}\mathbf{q}_t|$. The default value 2.0 is used if the read value of kappa is not positive. The variable is real.

4.8.7 maxdalpha

In adaptive quadrature, the maximum permitted change during iterations with the full data in the elements of the approximation to the location of the maximum of the conditional density of θ_i given $\mathbf{X}_i$ and $\mathbf{Z}_i$. The variable is real. The default value 3.0 of maxdalpha is used if the read value is not positive.

4.8.8 tau

For a regular iteration t, the smallest permitted ratio of $\alpha_{t(\nu+1)}/\alpha_{t\nu}$ for $\nu \geq 0$. The variable is real. The default value 0.1 is used if the read value of tau is not positive or is at least 1.

4.8.9 tol

The convergence criterion for iterations. Iterations terminate once $\ell_V(\boldsymbol{\gamma}_{t+1}) - \ell_V(\boldsymbol{\gamma}_t)$ is less than tol times $\ell_V(\boldsymbol{\gamma}_t)$. The variable is real. The default value 0.00001 is used whenever the input value of tol is not positive. In listeningacc.txt, tol is set to 10^{-8}, and the quadrature specification in quadsize uses 10 points rather than the default of 5 points. Results are in listeningacc.csv. The difference between results in listening.csv obtained with default settings is clearly quite small.

4.8.10 tola

For regular iterations, the convergence criterion for approximation of the location of the maximum of the posterior density of $\boldsymbol{\theta}_i$ given $\mathbf{X}_i$ and $\mathbf{Z}_i$. This criterion is only relevant for the case in which a model is used in which $\boldsymbol{\theta}_i$ has a multivariate normal distribution. The search stops for examinee i and iteration t once the change of $\mathcal{L}_i$ is less than tola. The variable is real, and the default value of 0.0001 is used if the read value of tola is not positive.

4.8.11 tolres

Tolerance for adjusted residuals (Haberman, 2009). Due to approximations errors encountered in use of iterative algorithms, adjusted residuals are not meaningful if the denominator used in their calculation is very small. If the estimated variance of a residual is less than tolres times the estimated variance of the corresponding observation or if the estimated variance of the observation is 0, then the adjusted residual is reported to be 0. The default value 0.01 is used if the input value of tolres is not positive.

4.8.12 tolsing

The criterion for $-\nabla^2 \ell_V(\boldsymbol{\gamma}_t)$ to be regarded as positive definite at regular iteration $t \geq 0$. The constant tolsing is real. If tolsing is not positive, then the default value 0.000000001 is used. The criterion for positive definiteness is that a Cholesky decomposition of $-\nabla^2 \ell(\boldsymbol{\gamma}_t)$ can be computed and that, for each positive integer c no greater than C, the square of row c and column c of the decomposition is greater than row c and column c of $-\nabla^2 \ell(\boldsymbol{\gamma}_t)$. This criterion is also applied to $\boldsymbol{\Phi}_V(\boldsymbol{\gamma}_t)$.

4.9 dimension

This namelist group specifies the dimensions K and D and provides information concerning $\mathbf{A}$. The group contains the following variables:

- dimlatin
- dimlatout
- custom

4.9.1 dimlatin

The dimension K of the latent vectors $\boldsymbol{\theta}_i$. The variable is an integer. Any read value less than 1 is changed to 1, and the default value is 1. Thus the default value is used in listening1.txt, but the value 4 is used in fourskill.txt.

4.9.2 dimlatout

The dimension D of the transformed latent vectors $\mathbf{A}\boldsymbol{\theta}_i$. The variable is an integer. Any read value less than 1 is changed to 1, and the default value is 1. As in the examples for dimlatin, the default value is used in listening1.txt, but the value 4 is used in fourskill.txt.

4.9.3 custom

A logical variable with .FALSE. as its default value. The variable is .TRUE. if, and only if, the matrix $\mathbf{A}$ does not assume its default value. In the default case, for $1 \leq d \leq D$ and $1 \leq k \leq K$, Row d and Column k of $\mathbf{A}$ is 1 if, and only if, $k = d + K - D$, $D < K$ and $k \leq K - D$, or $D > K$

and $d \le D - K$. Thus for $K = D$, **A** is the identity matrix. The default identity matrix applies in listening1.txt and fourskill.txt. In the restricted bifactor model in fourskillbi.txt, **A** has the default value for $K = 5$ and $D = 4$. All elements in the first column of **A** are one. The last four columns of **A** form the identity matrix.

4.10 linearspec

If custom of dimension is .TRUE., then the next namelist record is for the group linearspec with the following variable:

- lin_tran

4.10.1 lin_tran

The variable lin_tran is a real D by K array with the same default specifications as **A** has in dimension. For instance, in threefact.txt, an analysis is conducted with three factors and four skills. The general factor applies equally to all skills. The productive factor contrasts speaking and writing against listening and reading. The oral factor contrasts listening and speaking against reading and writing.

4.11 quadsize

This namelist provides the basic specification of quadrature points. The following variables are used:

- dimsize
- nquad
- cross
- equalpoints
- even
- fullgrid
- gausshermite

- grid
- normal
- simplex
- pspread

4.11.1 dimsize

This integer variable provides the number Q_k of quadrature points for each k from 1 to K if a grid is used and the value Q_k is constant for $1 \le k \le K$. This number is reduced to 30 if Gauss–Hermite quadrature is used and the input record has dimsize greater than 30. The default value is 5, and the default value is used if the read value is less than 2. For example, in fourskill.txt, the choice of dimsize equals 3 implies that each Q_k is 3 for k from 1 to 4.

4.11.2 nquad

This integer variable is the number Q of quadrature points. The default value is 0; however, in simplex quadrature, nquad is set to $K+1$ no matter what the namelist record contains, while in cross-polytope quadrature, nquad is set to $2K$ no matter what the namelist record contains. In the case of a grid, nquad is set to the product Q^* of the number Q_k of quadrature points in each dimension k for $1 \le k \le K$ if a full grid is specified or if the read value of nquad is less than 2 or greater than Q^*. If a grid is not used and neither simplex nor cross-polytope quadrature is used, then nquad must be an integer greater than 1. If this condition on nquad is not satisfied, then the program terminates with an error message. The file threefact19.txt provides an example in which nquad is set to 19 as in section 3.10.

4.11.3 cross

This logical variable has default value .FALSE., and the value is .TRUE. if, and only if, cross-polytope quadrature is used. This choice is considered in fourskillcr.txt. One may compare the corresponding results in fourskillcr.csv to those in fourskill.csv. It appears that in this example, there is some appreciable loss in use of the cross-polytope approach.

4.11.4 equalpoints

This logical variable has a default value of .TRUE., and is .TRUE. if, and only if, all Q_k are equal. An example with this variable .FALSE. is provided in fourskillmew.txt for a polytomous latent vector with fewer values for the writing factor than for the other factors. Results are in fourskillmew.csv.

4.11.5 even

This logical variable, which has a default value of .FALSE., is .TRUE. if quadrature points are based on a grid and if evenly spaced quadrature points are used. The variable is .FALSE. otherwise. If a grid is not used, then .FALSE. becomes the value of even no matter how this variable was specified in the namelist group. The use of even=.TRUE. is illustrated in fourskillme.txt, which is a polytomous analogue of fourskill.txt with four points per dimension. For results, see fourskillme.csv. Note that in terms of estimated expected penalty per item, the polytomous model used here performs a bit less well than does the multivariate normal model of fourskill.csv; however, differences are relatively small.

4.11.6 fullgrid

This logical variable, which has a default value of .TRUE., is .TRUE. if, and only if, a full grid is used, so that quadrature rules are defined as in section 3.5. In threefact19.txt, .FALSE. is the value of fullgrid, so that the quadrature points of section 3.10 can be used. Results are in threefact19.csv. They may be compared with results in threefact.csv, where a full grid was used. Differences in estimates, log-likelihoods, and information measures are small but not negligible.

4.11.7 gausshermite

This logical variable, which has a default value .TRUE., is .FALSE. if Gauss–Hermite quadrature is not used and .TRUE. otherwise. Whatever value is read in the namelist input, the variable is set to .FALSE. if .FALSE. is the value of grid or if .TRUE. is the value of even.

4.11.8 grid

This logical variable, which has default value .TRUE., is .TRUE. if, and only if, quadrature points are selected from the possible quadrature points associated with a product rule (section 3.5).

For consistency, grid is set to .FALSE. if .TRUE. is the value of either cross or simplex.

4.11.9 normal

This logical variable, which has default value of .TRUE., is .TRUE. if the latent vectors have normal distributions. Otherwise, latent vectors are polytomous and have values determined by the quadrature points used. For example, normal is .FALSE. in the polytomous analogue fourskillm.txt to fourskill.txt. Results found in fourskillm.csv are similar but a bit less satisfactory than the corresponding normal results in fourskill.csv. Note the modest increase in the estimated expected log penalty per item and in the corresponding Akaike and Gilula–Haberman measures.

4.11.10 simplex

This logical variable, which has default value .FALSE., is .TRUE. only if simplex quadrature is used. The indicator is set to .FALSE. if cross-polytope quadrature is specified by cross. This option is used in fourskillsi.txt. Results are in fourskillsi.csv. They are appreciably different than results in fourskill.csv, although they are remarkably similar given that simplex quadrature only involves 5 quadrature points rather than the 81 used in fourskill.csv.

4.11.11 pspread

This real variable determines the range of the quadrature points for even spacing. The range defined by pspread is used if the variable is positive. Otherwise, the range is determined by the program. For an example with pspread specified, see fourskillme.txt. The specification of 4.5 together with the use of dimsize equal to 4 leads to points at −2.25, −0.75, 0.75, and 2.25.

4.12 nquadperdim

If grid is .TRUE. and if equalpoints is .FALSE. in quadsize, then the Q_k, $1 \leq k \leq K$, are obtained from nquadperdim. This namelist group includes the following variable:

- Q

4.12.1 Q

The variable Q is an integer array with values Q_k, $1 \leq k \leq K$. Any read value of Q_k less than 2 is changed to the default value of 5. If Gauss–Hermite quadrature is used, then any read value of Q_k greater than 30 is changed to 30. This namelist appears in fourskillmew.txt, where $Q_k = 4$ for $k < 4$ and $Q_4 = 3$. Results in fourskillmew.csv are quite similar to those in fourskillme.csv obtained with $Q_k = 4$ for $1 \le k \le 4$.

4.13 griddata

This namelist group is read if fullgrid is .FALSE. and grid is .TRUE. in quadsize. An example is threefact19.txt. The following variables are used:

- coords
- cw

4.13.1 coords

The variable is an integer array with K rows and Q columns. Note that K is dimlatin in dimension and Q is nquad in quadsize. Row k and column q is i_{kq} for $1 \le k \le K$ and $1 \le q \le Q$. Specifications are based on an ordering of the sets $\mathcal{T}_k$ of section 3.5 that specify possible values of element k of a quadrature vector. Let $\mathcal{T}_k$ consist of the real numbers $\mathcal{M}_{ik}$, $1 \le i \le Q_k$, where $\mathcal{M}_{ik}$, $1 \le i \le Q_k$ is increasing in i. The set $\mathcal{Q}$ is determined by Q index vectors $\mathbf{i}_q$ of dimension K, $1 \le q \le Q$. Element k of $\mathbf{i}_q$ is i_{kq}, and $\mathcal{Q}$ contains the vectors $\mathcal{U}_q$, $1 \le q \le Q$, where element k of $\mathcal{U}_q$ is $\mathcal{U}_{kq} = \mathcal{M}_{i_{kq}k}$ for $1 \le k \le K$. The integer array coords of dimension K by Q has row k and column q equal to i_{kq}. The default values of $\mathbf{i}_q$, $1 \le q \le Q$, satisfy the constraint that $\mathbf{i}_1$ is the K-dimensional array with all elements 1. For $1 \le q < Q$, $\mathbf{i}_{q+1}$ is obtained by $\mathbf{i}_q$ by the rule that if $k(q)$ is the smallest positive integer with $i_{k(q)q} < Q_k$, then $i_{k(q+1)} = 1$ for $k < k(q)$, $i_{k(q)(q+1)} = i_{k(q)q} + 1$, and $i_{k(q+1)} = i_{kq}$ for $k > k(q)$. The program stops with an error message if any read i_{kq} is not positive or is greater than Q_k. In threefact19.txt, coords is defined as in section 3.10.

4.13.2 cw

The real array cw contains the Q elements $\mathcal{A}_q$, $1 \le q \le Q$. The default value of $\mathcal{A}_q$ is 1. The program stops with an error message if any read $\mathcal{A}(\mathcal{U}_q)$ is not positive. In threefact19.txt, values from section 3.10 have been multiplied by 36 to simplify input.

4.14 quaddim

If grid is .TRUE. but both even and gausshermite are .FALSE., then the namelist group quaddim is used to define $\mathcal{M}_{ik}$ and $\mathcal{Y}_k(\mathcal{M}_{ik})$ are read for $1 \le i \le Q_k$, $1 \le k \le K$. For $1 \le k \le K$, the namelist group quaddim is read. An example with evenly weighted and evenly spaced points is found in listeninge.txt, where a polytomous latent variable is used. Results in listeninge1.csv are rather similar to those in listening.csv. The following variables are in this namelist group:

- points
- weights

4.14.1 points

This real array of dimension Q_k contains $\mathcal{M}_{ik}$, $1 \le i \le Q_k$. Default values for $\mathcal{M}_{ik}$ are $q_{ik} = c_k[(2i - Q_k - 1)/2$, where $c_k = \{12/[(Q_k - 1)(Q_k + 1)]\}^{1/2}$. In listeninge.txt, seven points are evenly spaced from -1.5 to 1.5.

4.14.2 weights

This real array of dimension Q_k contains $\mathcal{Y}_k(\mathcal{M}_{ik})$ for $1 \le i \le Q_k$. Default values of $\mathcal{Y}_k(\mathcal{M}_{ik})$ are 1. The default selection is used in listeninge1.txt.

The default definitions result in $Q_k^{-1} \sum_{i=1}^{Q_k} q_{ik}^2 = 1$, so that (85) holds if $\mathcal{S}$ consists of all polynomial functions $\mathcal{P}$ on R^K such that, for each integer k, $1 \le k \le K$, and for element k of the argument of the polynomial, each term of $\mathcal{P}$ either is of odd degree or of degree no greater than 2.

4.15 quadrature

The namelist group quadrature is read if in quadsize, .FALSE. is the common value of cross, grid, and simplex. The group depends on the number Q of quadrature points specified by nquad and the dimension K specified by dimlatin. The following variables are in this group:

- vecpoints
- weights

4.15.1 vecpoints

The two-way real array vecpoints has dimension K by Q. If $\mathcal{Q}$ has members $\mathcal{M}_q$ with elements $\mathcal{M}_{kq}$, $1 \le k \le K$, $1 \le q \le Q$, then row k and column q of vecpoints is $\mathcal{M}_{kq}$. The default value of each element of vecpoints is 0. An illustration is provided in threeskill.txt, which considers a model for the Listening, Reading, and Speaking skills. Here there are 12 quadrature points, so that nquad is 12. In vecpoints, quadrature points $c_k\boldsymbol{\delta}_k + c_m\boldsymbol{\delta}_m$ are specified, where $|c_k| = |c_m| = (3/2)^{1/2}$ and $1 \le k < m \le K = 3$. As evident from weights, points are evenly weighted. This example is similar to the example in section 3.10, but (85) holds for $\mathcal{S}$ consisting of polynomials with terms with the degree of each element no greater than 3. Results in threeskill.csv may be compared with results in threeskill5.csv for the default quadrature. Results are similar, but differences are not negligible.

In listeningu.txt, a somewhat different case is considered. The set Ω of possible values of $\boldsymbol{\theta}_i$ consists of the pairs $(-1, 0)$, $(0, 0)$, and $(0, 1)$, and the 1 by 2 matrix $\mathbf{A}$ has both elements 1. As a result, $\mathbf{A}\boldsymbol{\theta}_i$ has a single element $\theta_{i+} = \theta_{i1} + \theta_{i2}$ with possible values -1, 0, and 1. Clearly there is a one-to-one correspondence between θ_{i*} and $\boldsymbol{\theta}_i$, so that the model specifications in allfactorspecs lead in effect to a 2PL model for Listening with a polytomous latent variable θ_{i+} with values -1, 0, and 1. Results are in listeningu.csv. Note that, in terms of the information measures PE, GH, and AK, they are relatively similar to those in listening.csv for the 2PL model with a standard normal distribution for the latent variable. The scaling of parameters is not the same, so that estimated item parameters are somewhat different. The two linear parameters are logarithms of ratios of probabilities. The first ratio is $P(\theta_{i+} = 0)/P(\theta_{i+} = -1)$. The second ratio is $P(\theta_{i+} = 1)/P(\theta_{i+} = 0)$. Knowledge of the two linear parameters determines the distribution of θ_{i+}.

4.15.2 weights

The real array weights has dimension Q. Element q of weights is $\mathcal{W}(\mathcal{M}_q)$. The default value of each element of weights is 1. This default value is used in threeskill.txt and listeningu.txt. In listeningbinw.txt, a nontrivial array weights is employed. In this example, which involves a special case of a restricted bifactor model, the $\mathcal{M}_{kq}$, $1 \le k \le K = 7$, $1 \le q \le Q = 1000$, q odd, are computer-generated pseudo-random numbers with independent standard normal distributions, and $\mathcal{M}_{k(q+1)} = -\mathcal{M}_{kq}$. The weights are obtained by an adjustment by minimum discriminant information (Haberman, 1984) to ensure that (85) holds for $\mathcal{S}$ including all polynomials $\mathcal{P}$ on the space R^7 of seven-dimensional vectors such that, for each integer k, $1 \leq k \leq 7$, and for element k of the argument of the polynomial, each term of $\mathcal{P}$ has odd degree or degree less than 4. Results in listeningbinw.csv are fairly close to those in listeningbi.csv obtained with Gauss–Hermite quadrature for $Q_k = 3$ for $1 \leq k \leq 7$. It should be noted that this example is relatively difficult for adaptive quadrature due to use of item sets with only five or six members. The weighting in listeningbinw.txt is quite important. The use of even weights and 1,000 sets of normal random numbers in listeningbin.txt is somewhat less successful for computation of parameter estimates, as is evident by comparison of listeningbin.csv, listeningbinw.csv, and listeningbi.csv. The basic issue is that in listeningbi.txt, (85) does not hold even for polynomials $\mathcal{P}$ of degree no greater than 2.

4.16 allfactorspecs

The namelist group allfactorspecs provides factor specifications that apply to all elements θ_{ik} of the latent vector $\boldsymbol{\theta}_i$ of person i. The following variables are in this group:

- factor_specs
- fix_diag
- fixquad
- independence
- nolin
- noquad

4.16.1 factor_specs

If factor_specs is .TRUE., then individual factor information is read with the namelist group factorspecs for each integer k from 1 to K. If factor_specs is .FALSE., the default value, then no individual factor information is read, and the factor names are 'Factork' for $1 \leq k \leq K$. In listening.txt, the only setting of allfactorspecs that is not the default value is factor_specs. This setting is used to name the factor in factorspecs.

4.16.2 fix_diag

In (12), the λ_{kk1} are $-1/2$ if fix_diag is .TRUE., the default value, and this constraint is not imposed if .FALSE. is the value of fix_diag. In the case of a normal distribution for the latent vectors, λ_{kk1} implies that 1 is the conditional variance of θ_{ik} given $\theta_{ik'}$, $k' \neq k$, and the predictors $Z_{iu} = \delta_{u1}$, $1 \leq u \leq U$. Thus, in the normal case, θ_{ik} has variance 1 if $K = 1$ and $U = 1$, while λ_{121} is the correlation coefficient of θ_{1i} and θ_{2i} if $K = 2$ and $U = 1$. A somewhat nontrivial use of fix_diag equals .FALSE. is found in fourskillbi.txt in the case of a restricted bifactor model. In this example, this selection of fix_diag is made because it applies to all but one of the $K = 5$ elements of $\boldsymbol{\theta}_i$. A related but different case is considered in listeningbinw.txt, where a restricted bifactor model is employed to treat item sets. The variances associated with the item sets are assumed equal but not necessarily the same as the variance for the Listening factor.

4.16.3 fixquad

If the logical variable fixquad is .TRUE., its default value, then $\lambda_{kk'u} = 0$ for $u > 1$. Thus, in the normal case, the conditional covariance matrix of the $\boldsymbol{\theta}_i$ given $\mathbf{Z}_i$ is constant for all examinees i. If fixquad is .FALSE., then $\lambda_{kk'u}$ is not assumed to be 0 for $u > 1$. In Four3Cn25twogq.txt, fixquad is .FALSE., so that the means and covariance matrices of the $\boldsymbol{\theta}_i$ both depend on the group to which the examinee belongs. The results in Four3Cn25twogq.csv suggest that the gain in model fit is quite limited relative to the results in Four3Cn25twog.csv, in which the covariance matrix of the $\boldsymbol{\theta}_i$ is not affected by group membership. The added estimates of the $\lambda_{kk'2}$ do appear to be nonzero; however, they are quite small, and the changes in information measures are also quite small.

4.16.4 independence

If the logical variable independence is .TRUE., then $\lambda_{kk'u} = 0$ for $k \neq k'$ and $1 \leq u \leq U$, so that, conditional on the predictors $\mathbf{Z}_i$, the θ_{ik}, $1 \leq k \leq K$, are independent. If independence is .FALSE., its default value, then the elements θ_{ik}, $1 \leq k \leq K$, are not conditionally independent given the $\mathbf{Z}_i$. In the restricted bifactor model in fourskillbi.txt, .TRUE. is the value of independence.

4.16.5 nolin

If the logical variable nolin is .TRUE., its default value, then $\psi_{k1} = 0$ for $1 \leq k \leq K$. In the normal case, this condition implies that the conditional expectation of each θ_{ik} is 0 given that $Z_{iu} = \delta_{u1}$ for $1 \leq u \leq U$. If nolin is .FALSE., then the ψ_{k1} are not set to 0 for $1 \leq k \leq K$. This case arises in the constrained case of a 2PL model in listeningc.txt. In the normal case, nolin equals .FALSE. corresponds to a mean of $\boldsymbol{\theta}_i$ that need not be $\mathbf{0}_K$. Thus, in listeningc.txt, the mean of the latent variable is not set to 0. Instead, linear constraints are imposed on item parameters. In listeningu.txt, nolin equals .FALSE. corresponds to the use of the linear parameters discussed in vecpoints.

4.16.6 noquad

If the logical variable noquad is .TRUE., then the λ_{kku} are all 0. If noquad is .FALSE., its default value, then λ_{kku} need not be 0. This condition cannot apply in the normal case, but it can be employed in the polytomous case for the construction of quite general models. In listeningu.txt, the use of noquad equals .TRUE. permits use of the linear parameters discussed in vecpoints to define the distribution of θ_{i+}.

4.17 factorspecs

If the logical variable factorspecs is .TRUE. in allfactorspecs, then individual factor specifications are provided by the K namelist records with group name factorspecs. The kth namelist record corresponds to factor k, $1 \leq k \leq K$. Default values are taken from allfactorspecs when variable names are common to the namelist record factorspecs and the namelist record allfactorspecs. The following variables are in the namelist group:

- factor_name
- fix_diag
- independence

4.17.1 factor_name

The character variable factor_name is the name of the factor. The name can contain up to 16 characters. In accordance with standard Fortran practice, the name should be enclosed by a pair of apostrophes or a pair of quotation marks. The default value 'Factor$k$' is used if no name is specified. For a simple example, see listening.txt, where 'Listening' is the name of the one factor. In fourskillbi.txt, factor names apply to a general factor and to four factors for specific skills.

4.17.2 fix_diag

If the logical variable fix_diag is .TRUE., its default value, then λ_{kk1} is $-1/2$. Otherwise λ_{kk1} is not fixed. This feature is used in threefact4.txt to establish a general factor with a fixed variance and productive and written factors with unknown variances. Another use is in listeningbinw.txt, where the variances for the item sets are fixed but not the variance associated with Listening.

4.17.3 independence

If the logical variable independence is .TRUE., then $\lambda_{kk'u} = 0$ for $k' \neq k$ and $1 \leq u \leq U$, so that θ_{ik} is conditionally independent given $\theta_{ik'}$, $k' \neq k$, and $\mathbf{Z}_i$.

4.18 allskillspecs

General specifications for the elements of the skill latent vector $\mathbf{A}\boldsymbol{\theta}_i$ are provided by the namelist group allskillspecs. This group includes the following variables:

- rasch_model
- rasch_slope_1
- repeat_names
- skill_specs

4.18.1 rasch_model

The logical variable rasch_model, which has default value .FALSE., determines whether a 1PL or PC model is used. If rasch_model is .TRUE., then $a_{dhj} - a_{d(h-1)j}$ is constant over skills d, categories h, and items j such that skill d in $D(j)$ applies to item j. For a simple case with rasch_model equal .TRUE., see listeningr.txt and listeningr.csv.

4.18.2 rasch_slope_1

The logical variable rasch_slope_1, which has default value .FALSE., determines whether the slope parameter associated with a skill is 1 under a 1PL model or PC model. If rasch_slope_1 is .TRUE., then $a_{dhj} - a_{d(h-1)j}$ is 1 for each skill d, category h from 1 to $H_j - 1$, and item j for which skill d in $D(j)$ applies to item j. If rasch_slope_1 is .TRUE., then .TRUE. is also the value of rasch_model. An example with rasch_slope_1 equal .TRUE. is found in listeningr1.txt. Output is in listeningr1.csv. As in this example, rasch_slope_1 equals .TRUE. is usually associated with fix_diag equals .FALSE. in allfactorspecs. Note that the information measures and item intercepts are the same in listeningr.csv and listeningr1.csv and the square of the estimated common item discrimination 1.152 in listeningr.csv corresponds to minus the inverse of 2 times the quadratic parameter -0.3775 in listeningr1.csv.

4.18.3 repeat_names

The logical variable repeat_names, which has default value .FALSE. if $D \neq K$ and default value .TRUE. if $D = K$, determines if skill names are derived from factor names. If repeat_names is .TRUE., then the name of skill d is set to the name of Factor d for $1 \leq d \leq \min(D, K)$. If repeat_names is .FALSE. and $1 \leq d \leq K$ or repeat_names is .TRUE., $D > K$, and $K < d \leq D$, then skill d has default name 'Skill$d$'. For example, in listening.txt, allskillspecs does not specify repeat_names, so that the default setting is used. Thus the one skill has the name 'Listening' that corresponds to the one factor name.

4.18.4 skill_specs

The logical variable skill_specs, which has default value .FALSE., determines if individual specifications are provided for each skill. If skill_specs is .TRUE., then a namelist record for the group skillspecs is provided for each of the D elements of $\mathbf{A}\boldsymbol{\theta}_i$. The value .TRUE. is used in fourskill.txt due to the use of the Rasch model for one skill out of four.

4.19 skillspecs

If skill_specs in the namelist group allskillspecs is .TRUE., then the control file must contain D namelist records for the group skillspecs. Record d, $1 \leq d \leq D$, corresponds to element d of $\mathbf{A}\boldsymbol{\theta}_i$. Each namelist group skillspecs includes the following variables:

- skill_name
- rasch_model
- rasch_slope_1

4.19.1 skill_name

This character variable provides the name of the skill for element d of $\mathbf{A}\boldsymbol{\theta}_i$. The name can contain up to 16 characters. In accordance with standard Fortran practice, the name should be enclosed by a pair of apostrophes or a pair of quotation marks. The default value is the value determined in allskillspecs. In fourskillbi.txt, the four skills correspond to the last four factors rather than the first four factors, so that their names are separately listed.

4.19.2 rasch_model

This logical variable is .TRUE. if, and only if, $a_{dhj} - a_{d(h-1)j}$ is assumed constant for skill d in $D(j)$ for integers h from 1 to $H_j - 1$. The default value is the value of rasch_model in allskillspecs. This option is used in fourskill.txt for the Writing skill due to the existence of only two items associated with this skill. Note that in fourskill.csv, only one slope parameter is associated with the two Writing items.

4.19.3 rasch_slope_1

This logical variable is .TRUE. if, and only if, $a_{dhj} - a_{d(h-1)j}$ is 1 for skill d in $D(j)$ for integers h from 1 to $H_j - 1$. If rasch_slope_1 is .TRUE., then .TRUE. is also the value of rasch_model. The default value is the value of rasch_slope_1 in allskillspecs.

4.20 allitemspecs

General item specifications are provided by the namelist group allitemspecs. This group includes the following variables:

- int_dim
- num_choices
- num_cat

- num_cat_obs
- slope_dim
- between_item
- cat_map
- constguess
- fixguess
- guessing
- item_specs
- nommod
- special_item_int
- special_item_slope
- set_guess

4.20.1 int_dim

The integer variable int_dim is the default number of parameters used to define item intercepts. If guessing is .TRUE., then int_dim is 2. If guessing is .FALSE., then the default value of int_dim is one less than num_cat. In addition, if guessing is .FALSE., then int_dim is set to 0 for a read value of int_dim is negative, and int_dim is set to one less than num_cat if int_dim is at least num_cat. In writing.txt, int_dim is 2, for a quadratic model is employed for the logarithm of the conditional probability given the latent variable that an item response has a specified value.

4.20.2 num_choices

The integer variable num_choices is the default number of multiple-choice categories for an item j. The variable is 0, its default value, or negative if the default number of multiple-choice items is not known or if the number of possible responses is not finite. In listeninggf.txt, which treats a 3PL model with a fixed guessing parameter, the value 4 is used. This choice leads to a guessing probability of 1/4. As evident from PE AK, and GH in listeninggf.csv and listening.csv, this choice of a fixed guessing parameter leads to a less satisfactory description of the data than does the simple 2PL model.

4.20.3 num_cat

The integer variable num_cat is the default number H_j of underlying item categories for each item j. If guessing is .FALSE. and num_cat is unspecified or any integer less than num_cat_obs, then num_cat is num_cat_obs. Note that num_cat_obs is at least 2. If guessing is .TRUE. and if num_cat_obs is 2, then num_cat is 4.

4.20.4 num_cat_obs

The integer variable num_cat_obs is the default number G_j of observed item categories for each item j. If num_cat_obs is unspecified or less than 2, then num_cat_obs is 2. For example, in speaking.txt, num_cat_obs is 4.

4.20.5 slope_dim

The integer array slope_dim has D elements. For $1 \le d \le D$, element d is the default number of slope parameters for each item for skill d. If element d is less than 0, then it is replaced by 0. If element d is greater than one less than num_cat, then it is replaced by one less than num_cat. The default value is 1 for each element of slope_dim.

4.20.6 between_item

The logical variable between_item is .TRUE., its default value, if a between-item model is used. The variable between_item is .FALSE. if a between-item model is not used. In the bifactor model in fourskillubi.txt, .FALSE. is the value of between_item.

4.20.7 cat_map

The logical variable cat_map is .TRUE. if the default values of the H_{xj} do not satisfy $H_{xj} = x$ for $1 \le x \le G_j - 1$. Otherwise, cat_map is .FALSE., its default value. If num_cat_obs and num_cat are the same or if guessing is .TRUE., then .FALSE. is the value of cat_map.

4.20.8 constguess

The logical variable constguess is .TRUE. if, and only if, a constant guessing parameter is used for all items associated with a 3PL model for a dichotomous item. The default value is .TRUE.; however, constguess is set to .FALSE. if fixguess is .TRUE., fixguess in itemspecs is .TRUE. for some item j, or if guessing is .FALSE. and .FALSE. is the value of guessing in itemspecs for each item j. In listeningggg.txt, a general 3PL model is defined, so that .FALSE. is the value of constguess. As evident from listeningggg.csv, this selection leads to substantial numerical problems and very poorly identified estimates. In listeningg.txt, the default value of constguess is used, and results in listeningg.csv are far more satisfactory.

4.20.9 fixguess

The logical variable fixguess is .TRUE. if a fixed guessing parameter is associated with a 3PL model for a dichotomous response. The variable is .FALSE., its default value, if the guessing parameter is not fixed or if .FALSE. is the value of guessing. This option is used in listeninggf.txt with num_choices equal to 4.

4.20.10 guessing

The logical variable guessing is .TRUE. if a 3PL model is used for dichotomous responses. The variable is .FALSE., its default value, otherwise. In listeningg.txt, guessing is .TRUE. and constguess is not specified, so that a 3PL model is applied to each item, and a constant guessing parameter is used. Results are in listeningg.csv.

4.20.11 item_specs

The logical variable item_specs is .TRUE. if individual item specifications are provided in itemspecs for each item j. If item_specs is .FALSE., its default value, then no individual item specifications are obtained, and the item name of item j is set to 'Item$j$'. If between_item is .TRUE. and if $D > 1$ (dimlatout exceeds 1 in dimension), then .TRUE. is the value of item_specs. In addition, .TRUE. is the value of item_specs if .TRUE. is the value of cat_map, special_item_int, or special_item_slope. In listening.txt, the default value of item_specs is used, so that individual item specifications are not read. In fourskill.txt, individual item specifications are required.

4.20.12 nommod

The logical variable nommod is .TRUE. if the default item specification for each item is a nominal model. The variable is .FALSE., its default value, if a nominal model is not specified for each item. For an illustration, see speakingm.txt and speakingm.csv.

4.20.13 special_item_int

If special_item_int is .TRUE., then a nonstandard parameterization is used for the item intercept. Thus the standard parameter definitions for a GPC model, nominal model, or 3PL model do not apply to the item. The indicator is .FALSE. if int_dim is 0. This option is used in writing.txt.

4.20.14 special_item_slope

If special_item_slope is .TRUE., then a nonstandard parameterization is used for the item slopes. If each element of slope_dim is 0, then .FALSE. is the value of special_item_slope. In writingsl.txt, .TRUE. is the value of special_item_slope. An attempt is made to analyze with the original scores of 0 and the integers 2 to 10. As evident from writingsl.csv, this attempt does not appear to have improved results relative to writing.csv.

4.20.15 set_guess

The real variable set_guess is the standard guessing parameter $\tau_{2j} - \tau_{0j}$ for each item j for which a 3PL is used. The value used for this guessing parameter is fixguess .TRUE., and the value is used as a starting value in iterations if fixguess is .FALSE. but .TRUE. is the value of guessing. If num_choices is greater than 1 and set_guess is not specified, then set_guess is minus the logarithm of 1 less than num_choices. This choice corresponds to guessing without any knowledge at all. If num_choices is not positive and set_guess is not specified, then set_guess is set to -1.

4.21 itemspecs

If item_specs is .TRUE. in allitemspecs, then a namelist record with group name itemspecs is read for each item from 1 to J. For each item, except for the variables item_name and skill_num, interpretations and default values are from the allitemspecs group; however, the values are now specific to the item. The following variables are used in the record for the namelist group for item j:

- item_name
- cat_map
- int_dim
- num_choices
- num_cat
- num_cat_obs
- skill_num
- slope_dim
- between_item
- fixguess
- guessing
- nommod
- special_item_int
- special_item_slope
- set_guess

4.21.1 item_name

The item name is a character variable specified by item_name. The name must have no more than 16 characters. The default value is 'Item$j$' for the jth item.

4.21.2 cat_map

The logical variable cat_map is .TRUE. if a special category mapping is required for the jth item. The variable is .FALSE. if no such mapping is required. The default value is the value of cat_map in allitemspecs. The variable is set to .FALSE. if num_cat_obs is the same as num_cat, so that $G_j = H_j$. The default category mapping has $H_{xj} = x$ for $0 \leq x < H_j$.

4.21.3 int_dim

The integer variable int_dim has default value $G_j - 1$ for item j unless int_dim in allitemspecs is not one less than the specified value of num_cat in allitemspecs. In this latter case, the default value of int_dim is the value of int_dim specified in allitemspecs. For example, in threefact.txt, int_dim assumes its default value for all items except for those associated with Speaking (skill_num=3) and Writing (skill_num=4). For the 34 Listening items (skill_num=1), num_cat_obs and num_cat are unspecified, so that the default value 2 is used for the number $G_j = H_j$ of underlying categories and the default value of $1 = 2 - 1$ is used for int_dim. For 39 of the 42 Reading items (skill_num=2), num_cat_obs and num_cat are unspecified, so that the default value 2 is used for the number G_j of underlying categories and the default value of $1 = 2 - 1$ is used for int_dim. For 3 Reading items, num_cat_obs is 3 and num_cat is unspecified, so that $G_j = H_j = 3$ and int_dim is set to $3 - 1 = 2$. In the case of the six Speaking items and two Writing items, int_dim is specified to be 2. If int_dim is specified to be negative, then the default value of int_dim is used. If guessing is .TRUE. and num_cat_obs is 2, then int_dim is set to 2.

4.21.4 num_choices

The integer variable num_choices is only relevant if a 3PL model is employed for item j. If num_choices is positive, then num_choices is the number of choices for item j. For example, this value is 4 if item j is a multiple-choice item with four choices. The default value of num_choices is provided by num_choices in allitemspecs.

4.21.5 num_cat

The integer variable num_cat is the number G_j of underlying categories for item j. The default is the value of num_cat in allitemspecs, and any value less than 2 is changed to 2. The value of num_cat is changed to num_cat_obs if the namelist input and the default values otherwise result in num_cat_obs exceeding num_cat. If guessing is .TRUE. and num_cat_obs is 2, then num_cat is set to 4. Note that if guessing is .TRUE. in allitemspecs and guessing is .FALSE. in itemspecs, then num_cat typically needs to be explicitly specified when it is not 4.

4.21.6 num_cat_obs

The integer variable num_cat_obs is the number H_j of observed categories for item j. The default is the value of num_cat_obs in allitemspecs, and any value less than 2 is changed to 2. In fourskill.txt, num_cat_obs is 3 for three Reading items (skill_num=2), num_cat_obs is 4 for six Speaking items (skill_num=3), and num_cat_obs is 10 for two Writing items (skill_num=4).

4.21.7 skill_num

The integer variable skill_num provides the skill number associated with item j if only one skill is associated with the item. The default value is 1. If skill_num has a read value less than 1, then the value is changed to 1, while the value is changed to D if skill_num has a read value greater than the number D of skills. In fourskill.txt, skill_name is 1 for 34 Listening items, 2 for 42 Reading items, 3 for 6 Speaking items, and 4 for two Writing items.

4.21.8 slope_dim

The integer array slope_dim has D elements. Element d, $1 \leq d \leq D$, provides the dimensions of the parameterization for the slope parameters a_{dhj}, $0 \leq h \leq H_j - 1$, for item j. The default value is provided by slope_dim in allitemspecs. If the input record has an element of slope_dim less than 0, then the element is changed to 0. If the input record has an element of slope_dim greater than $G_j - 1$ or if nommod is .TRUE., then the element is changed to $G_j - 1$. If between_item is .TRUE., then element d of slope_dim is changed to 0 whenever d is unequal to skill_num. In fourskillubi.txt, slope_dim is set for each item to produce a slope parameter for the general skill and a slope parameter for the specific skill. Results are in fourskillubi.csv. They may be compared to results in fourskillbi.csv for the restricted bifactor model. The statistics PE, AK, and GH indicate that the improvement in data description from the unrestricted model is somewhat limited in this case.

4.21.9 between_item

The logical variable between_item is defined as in allitemspecs in the sense that only one skill applies to item j. The default value is the value of between_item in allitemspecs.

4.21.10 fixguess

The logical variable fixguess is .TRUE. if a 3PL model is used for item j with a fixed guessing parameter. Otherwise, .FALSE. is the value of fixguess. The default value of fixguess is the value of fixguess in allitemspecs. The variable fixguess is set to .FALSE. if .FALSE. is the value of guessing.

4.21.11 guessing

The logical variable guessing is .TRUE. if a 3PL model is used for item j. Otherwise, .FALSE. is the value of guessing. The default value of guessing is the value of guessing in allitemspecs. The variable fixguess is set to .FALSE. if .FALSE. is the value of guessing, if num_cat_obs is greater than 2, or if .TRUE. is the value of nommod.

4.21.12 nommod

The logical variable nommod is .TRUE. if a nominal model is used for item j. Otherwise, .FALSE. is the value of nommod. The default value of nommod is the value of nommod in allitemspecs. The variable fixguess is set to .FALSE. if .FALSE. is the value of guessing, if num_cat_obs is greater than 2, or if .TRUE. is the value of nommod.

4.21.13 special_item_int

The logical variable special_item_int is .TRUE. if a special definition of the parameterization is required for the item intercepts τ_{hj}, $0 \leq h \leq H_j - 1$. Otherwise, .FALSE. is the value of special_item_int. The default value is the value of special_item_int in allitemspecs. The value of special_item_int is .TRUE. for Speaking and Writing items in threefact.txt.

4.21.14 special_item_slope

The logical variable special_item_slope is .TRUE. if a special definition of the parameterization is required for the item slopes a_{dhj}, $1 \leq d \leq D$, $0 \leq h \leq H_j - 1$. Otherwise,

.FALSE. is the value of special_item_slope. The default value is the value of special_item_slope in allitemspecs. In Four1Am81.txt, special_item_slope is .TRUE. for the last two items, where the actual item scores are 0, 2, 3, 4, 5, 6, 7, 8, 9, and 10.

4.21.15 set_guess

The real variable set_guess is the guessing parameter $\tau_{2j} - \tau_{0j}$ for item j if a 3PL is used. The default value is the value of set_guess in allitemspecs. The value is used for this guessing parameter if fixguess is .TRUE., and the value is used as a starting value in iterations if fixguess is .FALSE. but .TRUE. is the value of guessing. If num_choices is greater than 1, set_guess is not specified, and set_guess is not specified in allitemspecs, then set_guess is minus the logarithm of 1 less than num_choices.

Additional item specifications are read if, for any item, .TRUE. is the value of cat_map, special_item_int, or special_item_slope. Each record for item j is read before any record for item j' is read for $j' > j$.

4.22 catspecs

If cat_map is .TRUE., then the namelist record is read for group catspecs. This group includes the following variable:

- cat_array

4.22.1 cat_array

The real array cat_array has size $H_j - 1$. The values of cat_array are the H_{xj}, $1 \leq x \leq H_j - 1$. The default value for element x of cat_array is x.

4.23 intspecs

If special_item_int is .TRUE., then the namelist group intspecs is read. The following variable is in this group:

- int_array

4.23.1 int_array

The real array int_array has dimension G_j by int_dim. This array specifies $\mathbf{T}^1_{\tau j}$. The default mapping corresponds to the customary mapping for a GPC model or nominal model if no guessing parameter is used and corresponds to the 3PL parameter if the guessing parameter is used. This option is employed in threefact.txt for intercepts for Speaking and Writing.

4.24 slopespecs

If special_item_slope is .TRUE., then the namelist record for the group slopespecs is read for $1 \leq d \leq D$ for any integer d such that element d of slope_dim is positive. The group has the following variable:

- slope_array

4.24.1 slope_array

The variable slope_array is a two-dimensional real array. The first dimension is G_j and the second dimension is element d of slope_dim. The array defines $\mathbf{T}^2_{ad}$. In Four1Am81.txt, int_array and slope_array are selected to reflect linear and quadratic terms based on the actual item scores of 0, 2, 3, 4, 5, 6, 7, 8, 9, and 10. Results are in Four1Am81.csv.

4.25 predictorname

The namelist group predictorname is used to specify names of predictors. The group includes the following variable:

- pred_name

4.25.1 pred_name

This variable is an array of character variables, each of length 16 characters. The length is the sum of npred and nexternal. The initial elements of the array correspond to predictors in the model. The last elements are external predictors not used in the model. The default name for the initial predictor name is ‘Constant’, and, for either npred greater than 1 or nexternal positive, the default name of predictor u is ‘Predictor $u - 1$’. For any example, see Four3Cn25twog.txt and Four3Cn25twog.csv, where a model for two groups is used.

4.26 designparameters

The namelist group designparameters provides alternative parameterizations for the parameter vector $\boldsymbol{\gamma}$. The following variables are in this namelist group:

- constdim
- dimdesign
- proport
- specialtrans

4.26.1 constdim

The integer variable constdim is the number V of linear constraints on $\boldsymbol{\gamma}$. The default value is 0, and any negative value is changed to 0. In listeningc.txt, constdim is 2, for constraints are imposed on the sum of the item discriminations and the sum of the item slopes.

4.26.2 dimdesign

The integer variable dimdesign is the value of the dimension C of $\boldsymbol{\gamma}$. If not specified, C is the value computed from the standard model calculations. Processing stops unless dimdesign is positive. For a very simple example, consider speakingn.txt and speakingn.csv. Here the actual model considered is that the six Speaking responses are independently and identically distributed random variables. For such a model, only three independent parameters are needed, so that dimdesign is 3.

4.26.3 proport

The logical variable proport is .TRUE., its default value, if the quadratic constraint matrix is proportional to the sample size. This variable is only relevant if dimdesign is positive. In listeningggcc.txt and listeningggcc.csv, this variable is set to .FALSE. to employ maximum posterior likelihood in which each parameter γ_c, $1 \leq c \leq C = 102$, has an independent prior normal distribution with mean 0 and variance 0.5. Here the γ_c parameters include 34 item intercepts, 34 logits of item guessing parameters, and 34 item discriminations.

4.26.4 specialtrans

If this logical variable is .TRUE., then custom parameter names and custom values of $\mathbf{T}^2$ and $\mathbf{o}^2$ are read. The default value of .FALSE. results in use of customary parameter names and values of $\mathbf{T}^2$ and $\mathbf{o}^2$. In speakingn.txt, .TRUE. is the value of specialtrans, for a special linear model must be constructed for the parameters in the GPC model.

4.27 designspecs

If specialtrans is .TRUE., then the namelist group designspecs specifies the special design to be used. The following variables are in designspecs:

- param_name
- offsettran
- transition

4.27.1 param_name

The character variable param_name can have up to 64 characters. In speakingn.txt, the parameter names represent the common ratios $\log[P(X_{ij} = k+1)/P(X_{ij} = k)]$, $1 \leq j \leq 6$, for k from 0 to 2.

4.27.2 offsettran

The real array offsettran has dimension C and is equal to $\mathbf{o}$ in (33). In speakingn.txt, offsettran is just an array of 26 elements, each of which is 0.

4.27.3 transition

The two-dimensional real array transition has dimension B by C. Note that the array is read in standard Fortran order, so that rows vary faster than columns. In speakingn.txt, transition is a 26 by 3 array with all elements 0 or 1. In column k, values of 1 correspond to the common differences $\tau_{kj} - \tau_{(k-1)j}$, $1 \leq j \leq 6$.

4.28 constraints

If in designparameters, constdim is positive, then linear constraints are specified by the namelist group constraints. The following variables are in the group:

- const_mat
- const_vec

4.28.1 const_mat

The variable const_mat is a real array with row dimension C and column dimension constdim. The default value of each element of const_mat is 0 if the row and column are different and 1 if the row and column are the same. The array provides the transpose $\mathbf{S}'$ of the matrix $\mathbf{S}$ of section 2.5. In listeningc.txt, constdim is 2 in designparameters and fix_diag and nolin in allfactorspecs are .FALSE., so that $C = 70$. The choice of $\mathbf{S}'$ corresponds to linear constraints on the sums $\sum_{j=1}^{34} \gamma_j$ and $\sum_{j=35}^{68} \gamma_j$, where $\gamma_j = \tau_{1j} - \tau_{0j}$ and $\gamma_{34+j} = a_{11j} - a_{10j}$ for $1 \le j \le 34$.

4.28.2 const_vec

The real array const_vec is the array with size constdim that is the vector $\mathbf{s}$ in section 2.5. The default element of const_vec is 0. For example, in listeningc.txt, const_vec is the real array with elements 0 and 34, so that the average value of $\tau_{1j} - \tau_{0j}$ is 0 and the average value of $a_{11j} - a_{10j}$ is 1. Results are found in listeningc.csv.

4.29 readgamma

The namelist group readgamma specifies initial values $\boldsymbol{\gamma}_0$ for the computation of estimated parameter vector $\hat{\boldsymbol{\gamma}}$. The single variable in the group is the following:

- gammas

4.29.1 gammas

The real array gammas of C elements provides starting values for computation of $\hat{\boldsymbol{\gamma}}$. Element c of gammas is the starting value for $\hat{\gamma}_c$. If gammas is unspecified, then the program produces its own crude starting values. For example, these default starting values are used in listening.txt. In listeningst.txt, the estimates from listening.csv are used as input. Naturally, results in listeningst.csv are very similar to those in listening.csv.

4.30 inputinformation

If the model employed assumes that $\boldsymbol{\theta}_i$ is normally distributed, so that normal is .TRUE. in quadsize, then the namelist group inputinformation is used to specify initial values for $\boldsymbol{\theta}_{im}$ and $\mathbf{L}_i$ for each observation i. The following variables are included in this namelist group:

- fileformat
- filename
- readalpha

An example of use of this namelist group can be seen in listeningb.txt, which uses output from listeninga.txt. In listening.txt, one has a typical case in which default values are used for this namelist group.

4.30.1 fileformat

This character variable specifies the file format. The variable has up to 256 characters. List-directed input is used if '*', the default value, is the value of fileformat. Each record includes the K elements of $\boldsymbol{\theta}_{im}$ and the K by K array $\bar{\mathbf{L}}_i$. In listeningb.txt, this variable is '*', so that input is list directed.

4.30.2 filename

This character variable has up to 256 characters, and 'alpha.txt' is its default value. The value is 'listeningalpha.csv' in listeningb.txt.

4.30.3 readalpha

This logical variable with default value .FALSE. is true if, and only if, $\boldsymbol{\theta}_{im}$ and $\bar{\mathbf{L}}_i$ are to be read for each observation number i. The specification for $\mathbf{L}_i$ involves a K by K matrix $\bar{\mathbf{L}}_i$ such that the element in row k and column $k' < k$ of the matrix is the element in row k and column k' of $\mathbf{L}_i$ divided by the element in row k' and column k' of $\mathbf{L}_i$ and the element in row $k' \leq k$ and column k of the matrix is the element in row k and column k' of $\mathbf{L}_i$ multiplied by the element in row k' and column k' of $\mathbf{L}_i$. The default initial value for each $\boldsymbol{\theta}_{im}$ is the K-dimensional vector with all elements 0 and the default initial value of $\mathbf{L}_i$ and $\bar{\mathbf{L}}_i$ is the K by K identity matrix. In listeningb.txt, .TRUE. is the value of the variable. Obviously, results in listeninga.csv are quite similar to those in listeningb.csv.

4.31 printprogress

The namelist group printprogress specifies printing of iteration results. The group includes the following logical variables:

- printprogstart
- printprogstartstd
- printprog
- printprogstd

A typical example with all default settings is found in listening.txt. The resulting iteration summaries can be found in listening.csv. For each stage, the summary includes the iteration number, the number of steps required within the iteration, and the log-likelihood at the end of the iteration. The summary can be used to indicate convergence problems or unusually slow speed of computation, an important feature in complex models for large data files.

4.31.1 printprogstart

This variable is .TRUE., its default value, if iteration progress for the preliminary stage is to be printed to a file. If only one stage exists, then this variable is ignored.

4.31.2 printprogstartstd

This variable is .TRUE., its default value, if iteration progress for the preliminary stage is to be printed to standard output. If only one stage exists, then this variable is ignored. Note that iteration progress can be sent both to a file and to standard output. This option is helpful for monitoring program progress.

4.31.3 printprog

This variable is .TRUE., its default value, if iteration progress for the main stage is to be printed to a file.

4.31.4 printprogstd

This variable is .TRUE., its default value, if iteration progress for the main stage is to be printed to standard output.

4.32 output

Basic output specifications are provided by the namelist group output. The variables in the group are all logical variables in which a value .TRUE. implies printing the desired output and .FALSE. implies not printing. When requested, adjusted residuals are provided via a slight variation of a procedure developed for one-dimensional latent vectors (Haberman, 2009). Output files have comma-separated values. Examples appear in the discussion of units. The following variables are used:

- printalpha
- printeap
- printeapskill
- printeapwt
- printent
- printgrad
- printmargin
- printmarginres
- printmarginwtsum
- printmarginwtsumres
- printmargin2

- printmargin2res
- printmp
- printparam
- printparamcov
- printparamcov_complex
- printparamcov_louis
- printparamcov_sandwich
- printpost
- printprob
- printpreditem
- printpredtitemres
- printrel
- printrelskill
- printrelwt
- printwtitem
- printwtitemres

4.32.1 printalpha

If the logical variable printalpha is .TRUE., then the Fortran unit specified by unitalpha in units is used to store comma-separated values for $\boldsymbol{\theta}_{im}$ and $\mathbf{L}_i$ that have the format used for input from filename of inputinformation. For example, in listeninga.txt, the output is sent to listeningalpha.csv. This output is read in listeningb.txt. If printalpha is .FALSE., its default value, then values of $\boldsymbol{\theta}_{im}$ and $\mathbf{L}_i$ are not stored in an output file.

4.32.2 printeap

If the logical variable printeap is .TRUE., then EAP vectors and corresponding conditional covariance matrices for the underlying latent vectors $\boldsymbol{\theta}_i$ are provided in comma-separated format. The output unit is specified by uniteap in units. The initial output record provides a title. The second record specifies column interpretations. Each subsequent record corresponds to an individual. Details concerning EAP output are determined in the namelist group eapoutput. Examples of use of printeap equal to .TRUE. are found in listening1.txt and fourskillle.txt. If printeap is .FALSE., its default value, then EAP vectors and corresponding conditional covariance matrices are not saved in an output file.

4.32.3 printeapskill

If the logical variable printeapskill is .TRUE., then EAP vectors and corresponding conditional covariance matrices are provided for the transformed latent vectors $\mathbf{A}\boldsymbol{\theta}_i$. The output unit is specified by uniteapskill in units. The format is essentially the same as that used for output when .TRUE. is the value of printeap. Details concerning EAP output are determined in the namelist group eapoutput. An example of printeapskill equals .TRUE. is found in threefacte.txt. Note the output in threefacteapskill.csv. If printeapskill is .FALSE., its default value, then the EAP output for skills is not saved in an output file.

4.32.4 printeapwt

If the logical variable printeapwt is .TRUE. and if dimwtsum is .TRUE. in the namelist group eapoutput, then EAP vectors and corresponding conditional covariance matrices are provided for a vector of weighted sums. The output unit is specified by uniteapwt in units. The output format is essentially the same as for the output that results if .TRUE. is the value of printeap. Details concerning EAP output are determined in the namelist groups eapoutput and weightedsum. If dimwtsum is .FALSE., its default value, then EAP output for weighted sums is not provided. An example of printeapwt equal to .TRUE. is found in listening1.txt. The corresponding output file is listeningeapwt.csv.

4.32.5 printent

If the logical variable printent is .TRUE., its default value, then an information summary is provided in the output file that corresponds to unitinfo. For a description of the output for this option, see unitinfo. If printent is .FALSE., then the information summary is not provided.

4.32.6 printgrad

If the logical variable printgrad is .TRUE., then gradient $\nabla\ell_i(\hat{\gamma})$ is provided for each person i, $1 \leq i \leq n$, in the file associated with unit unitgrad. If printgrad is .FALSE., its default value, then this output is not provided. For example, in listening1.txt, the gradients are found in listeninggrads.csv.

4.32.7 printmargin

If the logical variable printmargin is .TRUE., then a summary of marginal distributions of items is provided in the file corresponding to unit unitmargin. If printmargin is .FALSE., its default value, then this summary is not provided. See unitmargin for further details. In listening1.txt, the summary is sent to listeningmarg.csv.

4.32.8 printmarginres

If the logical variable printmarginres is .TRUE., then the summary of marginal distributions of items on unitmargin includes information on residuals. If printmarginres is .TRUE., then .TRUE. is also the value of printmargin. If .FALSE., the default value, is the value of printmarginres, then adjusted residuals are not supplied on unitmargin. For an example with printmarginres equals .TRUE., see unitmargin. In listening1.txt, the adjusted residuals are found in listeningmarg.csv.

4.32.9 printmarginwtsum

If the logical variable printmarginwtsum is .TRUE., then a summary of the marginal distribution of weighted sums of item scores is provided in the file associated with unitmarginwtsum. If printmarginwtsum is .FALSE., its default value, then the summary is not provided. In listening1.txt, the summary of the marginal distribution of the sum of the response scores is found in listeningmargwtsum.csv. In listening1.txt, the adjusted residuals are found in listeningmargwtsum.csv.

4.32.10 printmarginwtsumres

If the logical variable printmarginwtsumres is .TRUE., then the summary of marginal distributions of weighted sums provided in the output file associated with unitmarginwtsum includes information on residuals. If printmarginwtsumres is .TRUE., then .TRUE. is also the value of printmarginwtsum. If printmarginwtsumres is .FALSE., its default value, then information on residuals is not provided for marginal weighted sums. In listening1.txt, the adjusted residuals are found in listeningmargwtsum.csv.

4.32.11 printmargin2

If the logical variable printmargin2 is .TRUE., then a summary of observed and fitted marginal distributions of item pairs is provided in the file associated with unitmargin2. If printmargin2 is .FALSE., its default value, then the summary is not provided. In listening1.txt, the summaries are found in listeningmarg2.csv.

4.32.12 printmargin2res

If the logical variable printmargin2res is .TRUE., then the summary of marginal distributions of item pairs includes information on residuals. If the logical variable printmargin2res is .TRUE., then .TRUE. is also the value of printmargin2. If printmargin2res is .FALSE., its default value, then the adjusted residuals are not provided. In listening1.txt, the adjusted residuals are found in listeningmarg2.csv.

4.32.13 printmp

If the logical variable printmp is .TRUE., then maximum a posteriori estimates and estimated information matrices are provided for examinees in the file associated with unit unitmp. This command is only used if the model uses a normal distribution for the latent vector. If printmp is .FALSE., its default value, then these estimates are not provided. In listening1.txt, the estimates are found in listeningmap.csv.

4.32.14 printparam

If the logical variable printparam is .TRUE., its default value, then estimates of $\boldsymbol{\gamma}$ and corresponding estimated asymptotic standard errors are provided in the output file associated with unit unitparam. If printparam is .FALSE., then these estimates are not provided.

4.32.15 printparamcov

If the logical variable printparamcov is .TRUE., then the standard estimated asymptotic covariance matrix of $\hat{\boldsymbol{\gamma}}$ is provided in the output file associated with unit unitparamcov. If printparamcov is .FALSE., its default value, then this estimated asymptotic covariance matrix is not supplied. In listening1.txt, the estimates are found in listeningcov.csv.

4.32.16 printparamcov_complex

If the logical variable printparamcov_complex is .TRUE., then the output file associated with unit unitparamcov_complex is used to provide the estimated asymptotic covariance matrix of $\hat{\boldsymbol{\gamma}}$ based on complex sampling. If printparamcov_complex is .FALSE., its default value, then this estimated asymptotic covariance matrix is not supplied. In listeningcc.txt, the estimated covariance matrix is in listeningcovc.csv.

4.32.17 printparamcov_louis

If the logical variable printparamcov_louis is .TRUE., then the Louis estimated asymptotic covariance matrix of $\hat{\boldsymbol{\gamma}}$ is provided in the output file associated with unit unitparamcov_louis. If printparamcov_louis is .FALSE., its default value, then this estimated asymptotic covariance matrix is not supplied. In listeningcl.txt, the estimated covariance matrix is in listeningcovl.csv.

4.32.18 printparamcov_sandwich

If the logical variable printparamcov_sandwich is .TRUE., then the sandwich estimated asymptotic covariance matrix of $\hat{\boldsymbol{\gamma}}$ is provided in the output file associated with unit unitparamcov_sandwich. If printparamcov_sandwich is .FALSE., its default value, then this estimated asymptotic covariance matrix is not supplied. In listeningcs.txt, the estimated covariance matrix is in listeningcovs.csv.

4.32.19 printpost

If the logical variable printpost is .TRUE., then a posterior distribution of $\boldsymbol{\theta}_i$ is provided for each observation i in the file associated with unit unitpost. The posterior distribution is specified by the quadrature points and the weights used for that observation to compute the log-likelihood function. If printpost is .FALSE., its default value, then posterior distributions are not provided. In listening1.txt, the posterior distributions are found in listeningpost.csv.

4.32.20 printpreditem

If the logical variable printpreditem is .TRUE., then observed and fitted totals and averages over all examinees are obtained for products of category indicator functions for examinee i and predictors Z_{iu} for $u > 1$. If printpreditem is .FALSE., its default value, then these sums are not obtained. To illustrate the case of printpreditem and printpredtitemres set to .TRUE., see Four3Cn25twog.txt and Four3Cn25twogpreditem.csv. In this case, the predictor is an indicator for membership in Group 2 rather than Group 1, so that the observed average for an item is the fraction of observations with both a correct response to the item and membership in Group 2. Thus a positive residual indicates that, in Group 2, more examinees answered the item correctly than expected from the fitted model.

4.32.21 printpredtitemres

If the logical variable printpredtitemres is .TRUE., then residuals are obtained for observed and fitted totals and averages over all examinees for products of category indicator functions for examinee i and predictors Z_{iu} for $u > 1$. If printpredtitemres is .FALSE., its default value, then these residuals are not obtained. If printpredtitemres is .TRUE., then .TRUE. is also the value of printpreditem.

4.32.22 printprob

If the logical variable printpost is .TRUE., then the estimated marginal probability for the observed response for each observation is provided in the file specified by unit printprob. If printprob is .FALSE., its default value, then estimated marginal probabilities are not provided. In listening3.txt, the estimates are found in listeningprob.csv.

4.32.23 printrel

If the logical variable printrel is .TRUE., then reliability coefficients for the elements $\hat{\theta}_{ik}$, $1 \le k \le K$, of the EAP $\hat{\boldsymbol{\theta}}_i$ of the underlying latent vector $\boldsymbol{\theta}_i$ are provided in the file corresponding to unit unitrel. Output also includes the estimated covariance matrices of $\hat{\boldsymbol{\theta}}_i$, $\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i$, and $\boldsymbol{\theta}_i$. If the logical variable printrel is .FALSE., its default value, then these estimated reliability coefficients and covariance matrices are not provided. In listening1.txt, the estimates are found in listening.csv.

4.32.24 printrelskill

If the logical variable printrelskill is .TRUE., then reliability coefficients for the elements of the EAP $\mathbf{A}\hat{\boldsymbol{\theta}}_i$ of the transformed latent vector $\mathbf{A}\boldsymbol{\theta}_i$ are provided in the file corresponding to unit unitrelskill. Output also includes estimated covariance matrices of $\mathbf{A}\hat{\boldsymbol{\theta}}_i$, $\mathbf{A}\boldsymbol{\theta}_i - \mathbf{A}\hat{\boldsymbol{\theta}}_i$, and $\mathbf{A}\boldsymbol{\theta}_i$. If the logical variable printrelskill is .FALSE., its default value, then these estimated reliability coefficients and covariance matrices are not provided. In threefacte.txt, the estimates are found in threefact.csv.

4.32.25 printrelwt

If the logical variable printrelwt is .TRUE., then reliability coefficients for the elements of the EAP $\widehat{\mathbf{TS}}_i$ of the expected weighted sum $\mathbf{TS}_i$ given the latent vector $\boldsymbol{\theta}_i$ and the covariate vector $\mathbf{Z}_i$ are provided in the file corresponding to unit unitrelwt. Output also includes estimated covariance matrices for $\widehat{\mathbf{TS}}_i$, $\widetilde{\mathbf{TS}}_i - \widehat{\mathbf{TS}}_i$, and $\widetilde{\mathbf{TS}}_i$. If the logical variable printrelwt is .FALSE., its default value, then these estimated reliability coefficients and covariance matrices are not provided. In listening1.txt, the estimates are found in listening.csv. In this example, $\mathbf{TS}_i$ is the sum score for the test for examinee i, and $\widetilde{\mathbf{TS}}_i$ is the test characteristic function at θ_{i1} for the sum score for the Listening test.

4.32.26 printwtitem

If the logical variable printwtitem is .TRUE., then observed and fitted totals and averages over all examinees are obtained for products of category indicator functions for examinee i and weighted sums defined in readweight. If printwtitem is .FALSE., its default value, then these sums are not obtained. To illustrate the case of printwtitem and printwtitemres set to .TRUE., see listening6.txt, listeningr.txt, listeningwtitem.csv, and listeningrwtitem.csv.

4.32.27 printwtitemres

If the logical variable printwtitem is .TRUE., then residuals are obtained for observed and fitted totals and averages over all examinees for products of category indicator functions for examinee i and weighted sums defined in readweight. If printwtitem is .FALSE., its default value, then these residuals are not obtained. If printwtitemres is .TRUE., then .TRUE. is also the value of printwtitem.

4.33 eapoutput

If printeap, printeapskill, printeapwt, printrel, printrelskill, or printrelwt is .TRUE., then the namelist group eapoutput specifies characteristics of EAP estimates to be computed. The group has the following members:

- dimwtsum
- eap_mask
- alt_beta

4.33.1 dimwtsum

The integer variable dimwtsum provides the dimension DS of the weighted sum $\mathbf{TS}_i$. If dimwtsum is not positive, then no weighted sum is considered. The default value of dimwtsum is 0. In listening2.txt, a single weighted sum is used, so that dimwtsum is 1. This weighted sum is the sum of the item scores for the Listening section. It is used in reliability estimation and in determination of EAP values. See listening2.csv and listeningeapwt.csv.

4.33.2 eap_mask

The logical array eap_mask has J elements. If element j of eap_mask is .TRUE., its default value, then response X_{ij} is used to compute the EAP. If element j of element eap_mask is .FALSE., then X_{ij} is not employed to compute EAP values for observation i. This option is sometimes relevant when data include both operational sections and external anchors, and EAP information is desired for the operational items. In listening4.txt, EAP estimates are based on the first 28 of the 34 items. Results are in listening4.csv and listening4eap.csv.

4.33.3 alt_beta

The real array alt_beta has the same dimension of $\hat{\boldsymbol{\beta}}$. It can be employed to change values of $\hat{\boldsymbol{\beta}}$. For each element of alt_beta, the default value is the corresponding element of $\hat{\boldsymbol{\beta}}$. In Four3Cn25twog2.txt, EAP estimates are obtained as if all examinees were from the first group. In Four3Cn25twog.txt, ordinary EAP estimates are obtained. This issue can arise from fairness consideration in an examination. It is not usually appropriate to give two examinees with identical responses different scores because they are in different groups. Results are in Four3Cn25twog2.csv and Four3Cn25twog2eap.csv for the EAP estimates that ignore group differences and in Four3Cn25twog.csv and Four3Cn25twogeap.csv for the conventional EAP estimates. The results are obviously the same for members of Group 1. Differences for members of Group 2 are quite small.

4.34 weightedsum

If DS is positive and if printeapwt or printrelwt is .TRUE., then an input record for the namelist group weightedsum is read for each dimension d from 1 to DS. This group has the following two variables:

- weight_name
- weight_sum

4.34.1 weight_name

This character variable of length 16 provides a name for the dth weighted sum. The default value for weight_name is the name of skill d (see skill_name) if either d is no greater than DS and DS $\leq D$ or d is not greater than D and DS $> D$. If DS is 1 and $D > 1$, then the default value is 'Total', while for DS greater than 1 but less than D, the default value for d equals DS is 'Remainder'. If DS is $D + 1$ and $D > 1$, then the default value for $d = D + 1$ is 'Total'. If DS exceeds $D + 1$ and $d > D + 1$ or DS is $D + 1$, $D = 1$, and $d > D$, then the default value is 'Sum_d'. In listening2.txt, weightname is 'Listening_sum'.

4.34.2 weight_sum

The real array weight_sum has dimension $\sum_{j\in J} H_j$. Element $\sum_{j'=1}^{j} H_{j'} + (h - H_j)$ corresponds to element d of $\mathbf{IS}_j(h-1)$ for $1 \le h \le H_j$ and $1 \le j \le J$. The default value for element $\sum_{j'=1}^{j} H_{j'} + (h - H_j)$ of the array is g if h is in $\mathcal{H}_{gj}$ for $0 \le g \le G_j - 1$, the number of slope parameters for skill d is positive, and either d is no greater than DS and DS $\le D$ or d is not greater than D and DS $> D$. If DS is $D+1$ and $D > 1$, then the default value is g if h is in $\mathcal{H}_{gj}$. Otherwise, the default value is 0. In listening2.txt, the default value is used, so that the weighted sum is simply the sum of the Listening item scores. Note results in listening2.csv and listeningeapwt.csv. For a more complex case, consider listening5.txt, listening5.csv, and listeningeap5.csv. In this case, the Listening sum is divided into two components, the sum for the first half of the test and the sum for the second half of the test. Because reported reliability estimates are for expected sums given the latent variable, these estimates are quite similar for the two sums of scores for halves of the test and for the total sum. This situation would be quite different were reliability estimates from classical test theory computed for these three sums of item scores.

4.35 numberweights

If printmarginwtsum is .TRUE., then numberweights provides the number of weighted sums used for marginal distributions. This namelist group has the following element:

- numweights

4.35.1 numweights

The integer variable numweights is the number of weighted sums used for marginal distributions. The default value of numweights is 0, and any read negative value of numweights is changed to 0. If numweights is positive, then numweights namelist groups readweight are read. In listening1.txt, numweights is 1, so that one weighted sum is used.

4.36 readweight

The namelist group readweight has the following variables:

- weightname

- weight

4.36.1 weightname

The character variable weightname has length 16. Default values are determined as in the case of weight_name for the dth group, except that DS is replaced by numweights. In listening1.txt, weightname is the name 'Listening_sum' assigned to the sum of the item scores.

4.36.2 weight

The integer array weight has $\sum_{j=1}^{J} G_j$ elements w_{gjd}, $0 \le g \le G_j$, $1 \le j \le J$, and the sum $\sum_{j=1}^{J} w_{X_{ij}j}$ is considered for each examinee i such that item j is presented whenever w_{gjd} is not constant for $0 \le g \le G_j - 1$. The distribution of the weighted sum under the model is computed by use of a recursive function. The algorithm is closely related to a procedure of Lord and Wingersky (1984) for dichotomous items that was generalized by Thissen, Pommerich, Billeaud, and Williams (1995). In listening1.txt, the default option for weight applies, so that the sum of the item scores is computed. Results are found in listeningmargwtsum.csv. Both the marginal distribution and adjusted residuals are provided. An examination of the adjusted residuals shows some model deviation, although the absolute size of errors is relatively small.

4.37 Projected Additions

Additional summaries and associated residuals are planned that involve totals and averages of products of category indicators and either predicting variables or external variables. Summary statistics are also planned in which conditional and unconditional expectations are compared for functions of observed responses, predicting variables, external variables, and latent vectors.

Simplified procedures are planned for fixing specific parameter values and for imposition of linear constraints on parameters that are not required for model identification. For example, options are planned to permit specification of values for all item discriminations without resorting to designspecs. Currently one can readily specify that all item discriminations associated with a skill must be the same or must be 1; however, one cannot readily impose other restrictions on item discriminations.

Procedures are planned to automate dummy coding of predictors and to automate use of interactions of predictor variables.

Plans exist to compute a variety of functions of model parameters on request. For example, item difficulties and associated estimated asymptotic standard errors are planned. In addition, the estimated covariance matrix and correlation matrix of the underlying latent vector or the transformed latent vector are to be reported if requested.

To facilitate efficient computation when the dimension K is relatively large or when each examinee receives only a small fraction of the items, additional quadrature options are planned. Special attention will be given to hierarchical structures such as bifactor models (Gibbons et al., 2007; Gibbons & Hedeker, 1982).

Methods for model comparison and methods for selection of subsets of observations for analysis have not yet been implemented.

References

Adams, R. J., Wilson, M., & Wang, W. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement, 21*, 1–23.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*, 716–723.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), *Statistical theories of mental test scores* (pp. 395–479). Reading, MA: Addison-Wesley.

Bock, R. D. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika, 37*, 29–51.

Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika, 46*, 443–459.

Box, G. E. P., Hunter, J. S., & Hunter, W. G. (2005). *Statistics for experimenters: Design, innovation, and discovery* (2nd ed.). Hoboken, NJ: John Wiley.

Coxeter, H. S. M. (1963). *Regular polytopes* (2nd ed.). New York, NY: Macmillan.

Cramŕ, H. (1946). *Mathematical methods of statistics.* Princeton, NJ: Princeton University Press.

Davis, P. J., & Polonsky, I. (1965). Numerical interpolation, differentiation, and integration. In M. Abramowitz & I. A. Stegun (Eds.), *Handbook of mathematical functions* (pp. 875–924). New York, NY: Dover.

de Bruijn, N. G. (1970). *Asymptotic methods in analysis* (3rd ed.). New York, NY: Interscience.

Fischer, G. H. (1973). The linear logistic test model as an instrument in educational research. *Acta Psychologica, 37*, 359–374.

Fisher, R. A. (1925). Theory of statistical estimation. *Proceedings of the Cambridge Philosophical Society, 22*, 700–725.

Fuller, W. A. (2009). *Sampling statistics.* Hoboken, NJ: John Wiley.

Gibbons, R. D., Bock, R. D., Hedeker, D., Weiss, D. J., Segawa, E., Bhaumik, D. K., . . . , Stover, A. (2007). Full-information item bifactor analysis of graded response data. *Applied Psychological Measurement, 31*, 4–19.

Gibbons, R. D., & Hedeker, D. (1992). Full information item bi-factor analysis. *Psychometrika, 57*, 423–436.

Gilula, Z., & Haberman, S. J. (1994). Models for analyzing categorical panel data. *Journal of the American Statistical Association, 89*, 645–656.

Gilula, Z., & Haberman, S. J. (1995). Prediction functions for categorical panel data. *Annals of Statistics, 23*, 1130–1142.

Gilula, Z., & Haberman, S. J. (2001). Analysis of categorical response profiles by informative summaries. *Sociological Methodology, 31*, 129–187.

Golub, G. H., & Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of Computation, 23*, 221–230.

Haberman, S. J. (1973). Log-linear models for frequency data: Sufficient statistics and likelihood equations. *Annals of Statistics, 1*, 617–632.

Haberman, S. J. (1979). *Analysis of qualitative data: Vol. II. New developments.* New York, NY: Academic Press.

Haberman, S. J. (1984). Adjustment by minimum discriminant information. *Annals of Statistics, 12*, 971–988.

Haberman, S. J. (1988). A stabilized Newton–Raphson algorithm for log-linear models for frequency tables derived by indirect observation. *Sociological Methodology, 18*, 193–211.

Haberman, S. J. (1989). Concavity and estimation. *Annals of Statistics, 17*, 1631–1661.

Haberman, S. J. (1996). *Advanced statistics: Vol. 1. Description of populations.* New York, NY: Springer.

Haberman, S. J. (2006). *Adaptive quadrature for item response models* (Research Report No. RR-06-29). Princeton, NJ: Educational Testing Service.

Haberman, S. J. (2009). *Use of generalized residuals to examine goodness of fit of item response models* (Research Report No. RR-09-15). Princeton, NJ: Educational Testing Service.

Haberman, S. J., & Sinharay, S. (2010). Reporting of subscores using multidimensional item response theory. *Psychometrika, 75*, 209–227.

Haberman, S. J., & Sinharay, S. J. (in press). *Generalized residuals for general models for contingency tables with application to item response theory.* Princeton, NJ: Educational Testing Service.

Haberman, S. J., von Davier, M., & Lee, Y. (2008). *Comparison of multidimensional item response models: Multivariate normal ability distributions versus multivariate polytomous distributions* (Research Report No. RR-08-45). Princeton, NJ: Educational Testing Service.

Heinen, T. (1996). *Latent class and discrete latent trait models.* Thousand Oaks, CA: Sage.

Hochstrasser, U. W. (1965). Orthogonal polynomials. In M. Abramowitz & I. A. Stegun (Eds.), *Handbook of mathematical functions* (pp. 771–819). New York, NY: Dover.

Holland, P. W. (1990). The Dutch identity: A new tool for the study of item response models. *Psychometrika, 55*, 5–18.

Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 221–233). Berkeley: University of California Press.

Isserlis, L. (1918). On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika, 12*, 134–139.

Kantorovich, L. V., & Akilov, G. P. (1964). *Functional analysis in normed spaces.* Oxford, England: Pergamon Press.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems.* Hillside, NJ: Lawrence Erlbaum Associates.

Lord, F. M., & Wingersky, M. S. (1984). Comparison of IRT true-score and equipercentile observed-score "equatings." *Applied Psychological Measurement, 8*, 453–461.

Louis, T. (1982). Finding the observed information matrix when using the em algorithm. *Journal of the Royal Statistical Society, Series B, 44*, 226–233.

Masters, G. N. (1982). A Rasch model for partial credit scoring. *Psychometrika, 47*, 149–174.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). Boca Raton, FL: Chapman and Hall.

Mislevy, R. J., Johnson, E. G., & Muraki, E. (1992). Scaling procedures in NAEP. *Journal of Educational Statistics, 17*, 131–154.

Mosteller, F., & Wallace, D. (1964). *Inference and disputed authorship: The Federalist.* Reading, MA: Addison-Wesley.

Muraki, E. (1991). *Parscale: Parametric scaling of rating data.* Chicago, IL: Scientific Software.

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement, 16*, 159–176.

Naylor, J. C., & Smith, A. F. M. (1982). Applications of a method for the efficient computation of posterior distributions. *Applied Statistics, 31*, 214–225.

Ralston, A. (1965). *A first course in numerical analysis.* New York, NY: McGraw-Hill.

Rao, C. R. (1973). *Linear statistical inference and its applications* (2nd ed.). New York, NY: John Wiley.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Danish Institute for Educational Research.

Savage, L. (1971). Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association, 66*, 783–801.

Schilling, S., & Bock, R. D. (2005). High-dimensional maximum marginal likelihood item factor analysis by adaptive quadrature. *Psychometrika, 70*, 533–555.

Sinharay, S., Haberman, S. J., & Lee, Y.-H. (2011). When does scale anchoring work: A case study. *Journal of Educational Measurement, 48*, 61–80.

Thissen, D., Pommerich, M., Billeaud, K., & Williams, V. (1995). Item response theory for scores on tests including polytomous items with ordered responses. *Applied Psychological Measurement, 19*, 39–49.

von Davier, M. (2008). A general diagnostic model applied to language testing data. *British Journal of Mathematical and Statistical Psychology, 61*, 287–307.

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica, 48*, 817–830.

Wilkinson, J. H. (1963). *Rounding errors in algebraic processes.* Englewood Cliffs, NJ: Prentice Hall.