The BMIRT Toolkit

Lihua Yao

Defense Manpower Data Center

DoD Center Monterey Bay

Lihua.Yao.civ@mail.mil

January 21, 2016

Contents

Chapter 1

Introduction

The software suite is called the BMIRT Toolkit, where BMIRT stands for Bayesian multivariate item response theory; they can be downloaded at www.BMIRT.com. The BMIRT Toolkit consists of four different software programs: a calibration program (BMIRT II), a linking program (LinkMIRT), a simulation program for a multidimensional fixed form test (SimuMIRT), and a simulation program for a multidimensional computer adaptive test (SimuMCAT). Each software is described below. Even though they were started at a different time, however, more features are being added and updated for each of the software constantly and continuously. No programing skills are needed in using any of the software.

1.1 BMIRT(2003)

It is a multi-purpose program that uses Markov Chain Monte Carlo methods to conduct item calibrations and ability estimation in a multidimensional, multi-group item response theory (IRT) model framework. BMIRT II has extensive capabilities. Both confirmatory and exploratory item factor analysis are possible. The program can perform unidimensional or multidimensional calibrations. It can operate on a single group or multiple groups. It can fit dichotomous or polytomous models (along with mixed models), including the three-parameter logistic model, the two-parameter logistic model, the Rasch model, the generalized two-parameter partial credit model, the testlet model, the graded response model, and the higher-order IRT model. It also has the capability of fixing parameters for anchor items and estimating parameters for non-anchor items. The program can compute three types of ability estimates besides MCMC, maximum a posteriori estimates, expected a posterior, and maximum likelihood estimates. The program can compute both domain scores and overall scores. It also has the capability of computing test response functions, item information functions, test information functions, model fit statistics, and classification accuracy indices. Rate-effect model and its estimations and a procedure for DIF analysis through using BMIRT multidimensional multi-group feature were implemented in 2010; research papers regarding them were presented at NCME 2010, 2011 conferences.

1.2 LinkMIRT(2009)

It is a program that links two sets of item parameters in a multidimensional IRT (MIRT) framework. The software can implement the Stocking and Lord method, the mean/mean method, and the mean/sigma method. Linking by comment-person and by random equivalent-groups design were implemented recently in 2012 and the research study comparing them is under review in a book chapter.

1.3 SimuMIRT(2003)

It is a program that simulates multidimensional data (examinee ability and item responses) for a fixed form (i.e., paper and pencil) test, from a user-specified set of parameters. The rater-effect model was implemented in 2011.

1.4 SimuMCAT (2011)

It is a program that simulates a multidimensional computer adaptive test (MCAT). The user can select from five different MCAT item selection procedures (Volume, Kullback-Leibler information, Minimize the error variance of the linear combination, Minimum Angle, and Minimize the error variance of the composite score with the optimized weight). Two exposure control approaches are possible: the traditional Sympson-Hetter approach and a maximum exposure control approach. It is also possible to implement content constraints using the Priority Index method. Different stopping rules are implemented with fixed-length test and varying-length test. The user specifies true examinee ability, item pools, and item selection procedures, and the program outputs selected items with item responses and ability estimates. Bayesian and non-Bayesian methods can be specified by the user. The examinee's ability and item pools can also be created from the program by the user specified distributions. These features are discussed in Lihua's recent publications in Psychometrkia, Applied Psychological Measurement, and Journal of Educational Measurement in 2012.

1.5 GIPOOL(2014)

GIPOOL (Generate item Pool) is a program that generate optimized item pool based on expected information (Yao, 2014c). Wait for update.

1.6 Install Software

1.6.1 Java Run Time Environment

You need to install Java runtime environment from JRE from internet, then set your computer " properties/ Environment Variable ", add the path where the Java runtime environment are located. For example," C:/Program

Files/Java/j2re/bin".

1.6.2 Software

The software are ran using DOS commend by double click the ".bat " file that you created. For example, a file named *final.bat* contains a line: *BMIRT.bat sample.ctl sample.rwo sampleout* . Here *BMIRT.bat* is given using the complied library *lib*; *sample.ctl* and *sample.rwo* are input files; *sampleout* is the name for ouput files. Each software has a main library that you need to download at www.BMIRT.com and copy into each working folder. Download BMIRT.zip for calibration and the library is *lib*. Download LinkMIRT.zp and the library for linking is *MEQTlib*. Download SimuMIRT.zip and the library for simulation is *SimuRwolib*. Download SimuMCAT.zip and the library for CAT simulation is *lib*.

Chapter 2

Models

Please pay attention that the item parameters in the model discussed below are the output of software.

2.1 Three-parameter Logistic Model (M-3PL)

For a j, the probability of a correct response to item j for an examinee with ability $\vec{\theta}_i = (\theta_{i1}, \cdots, \theta_{iD})$ for the multidimensional three-parameter logistic (M-3PL; Reckase, 1997) model is:

$$P_{ij1} = P(x_{ij} = 1 \mid \vec{\theta}_i, \vec{\beta}_j) = \beta_{3j} + \frac{1 - \beta_{3j}}{1 + e^{(-\vec{\beta}_{2j} \odot \vec{\theta}_i^T + \beta_{1j})}}, \tag{2.1}$$

where

$x_{ij} = 0$ or 1 is the response of examinee i to item j.

$\vec{\beta}_{2j} = (\beta_{2j1}, \cdots, \beta_{2jD})$ is a vector of dimension D for item discrimination parameters.

β_{1j} is the scale difficulty parameter.

β_{3j} is the scale guessing parameter.

$\vec{\beta}_{2j} \odot \vec{\theta}_i^T = \sum_{l=1}^{D} \beta_{2jl}\theta_{il}$.

The parameters for jth item are

$$\vec{\beta}_j = (\vec{\beta}_{2j}, \beta_{1j}, \beta_{3j}), \tag{2.2}$$

The item parameters in BILOG are in different format; it is $1.7a(\theta - b)$.

2.2 Generalized Two-parameter Partial Credit Model (M-2PPC)

For a polytomous scored item j, the probability of a response $k-1$ to item j for an examinee with ability $\vec{\theta}_i$ is given by the multi-dimensional version of the partial credit model (M-2PPC; Yao & Schwarz, 2006) :

$$P_{ijk} = P(x_{ij} = k-1 \mid \vec{\theta}_i, \vec{\beta}_j) = \frac{e^{(k-1)\vec{\beta}_{2j}\odot\vec{\theta}_i^T - \sum_{t=1}^{k}\beta_{\delta_t j}}}{\sum_{m=1}^{K_j} e^{((m-1)\vec{\beta}_{2j}\odot\vec{\theta}_i^T - \sum_{t=1}^{m}\beta_{\delta_t j})}}, \tag{2.3}$$

where

$x_{ij} = 0, \cdots, K_j - 1$ is the response of examinee i to item j.

$\vec{\beta}_{2j} = (\beta_{2j1}, \cdots, \beta_{2jD})$ is a vector of dimension D for item discrimination parameters.

$\beta_{\delta_k j}$ for $k = 1, 2, \ldots, K_j$ are the threshold parameters, $\beta_{\delta_1 j} = 0$, and K_j is the number of response categories for the jth item.

The parameters for jth item are

$$\vec{\beta}_j = (\vec{\beta}_{2j}, \beta_{\delta_2 j}, \ldots, \beta_{\delta_{K_j} j}), \tag{2.4}$$

2.3 Testlet Model

Testlet-effect-2PPC/3PL model is a constrained M-2PPC/3PL model. This model essentially puts a constraint on the discrimination parameter within each testlet or cluster of inter-related items in a form of a constant. The discrimination parameter varies across testlets to account for the testlet effect. Suppose there are $D-1$ testlets for a test. Then the model can be D dimensional IRT model, and the discrimination parameters are

$$\vec{\beta}_{2j} = (\beta_{2j1}, \beta_{2j1}\gamma_1, \beta_{2j1}\gamma_2, \cdots, \beta_{2j1}\gamma_{D-1}) \tag{2.5}$$

where $\gamma = (\gamma_1, \cdots, \gamma_{D-1})$ are the variances of the testlet-effect parameters for the $D-1$ testlets. Within each testlet, the ratio of the item general discrimination (β_{2j1}) and the item testlet-effect discrimination is a constant, namely testlet-effect parameter γ_k, where $k \in \{1, \cdots, D-1\}$. The other item parameters (item difficulty/guessing or threshold) remain the same as the general MIRT model. For a testlet-effect model based on a common stimulus, each item belongs to only one testlet, i.e., the discrimination parameters for item j is $(\beta_{2j1}, \beta_{2j1}\gamma_{\delta_j})$, where $\delta_j \in \{1, 2, \cdots, D-1\}$. For multiple criteria scoring rubric application, each item may contribute to more than one testlet; the items are grouped according to the rubric of grammar, meaning and appropriateness for example. As in Li, Bolt, and Fu (2004) or DeMars (2006), the formulas presented here are consistent with those found in the existing testlet models by Bradlow, Wainer, and Wang (2007). For item j in kth testlet-effect,

$$\vec{\beta}_{2j} \odot \vec{\theta}_i^T = \beta_{2j1}\theta_{i1} + \beta_{2j1}\gamma_k\theta_{ik}, \tag{2.6}$$

and $\gamma_k\theta_{ik} \sim N(0, \gamma_k^2)$.

2.4 Multidimensional Graded Response Model (M-GR)

For a polytomous scored item j with response level/category K_j, the multidimensional graded response model (M-GR; Muraki & Carlson, 1993) is defined below:

First define cumulative response function for $k = 0, \cdots, K_j$ as:

$P^{\star}_{ijk} = 1$ for $k = 0$.

$P^{\star}_{ijk} = 0$ for $k = K_j$.

$P^{\star}_{ijk} = 1 - \frac{1}{1+e^{\vec{\beta}_{2j}\odot\vec{\theta}+\beta_{\delta_k j}}} = \frac{1}{1+e^{-\vec{\beta}_{2j}\odot\vec{\theta}-\beta_{\delta_k j}}}$ for $k = 1, \cdots, K_j - 1$.

The probability of having response $k-1$ with $k \in \{1, \cdots, K_j\}$ for item j is

$$P_{ijk} = P(x_{ij} = k-1 \mid \vec{\theta}_i, \vec{\beta}_j) = P^{\star}_{ijk-1} - P^{\star}_{ijk} \tag{2.7}$$

The parameters for jth item are

$$\vec{\beta}_j = (\vec{\beta}_{2j}, \beta_{\delta_1 j}, \ldots, \beta_{\delta_{K_j} j}), \tag{2.8}$$

Note that $\beta_{\delta_1 j} \leq \beta_{\delta_2 j} \leq \cdots \leq \beta_{\delta_{K_j-1} j}$

2.5 Higher-Order IRT Model

For this model, the first-order follows IRT model, which describes the item performance for a given domain ability. The second-order describes linear relations between domain abilities and overall abilities. The domain abilities are expressed as linear functions of the overall ability, $\theta_{il} = \lambda_l\theta_i + \eta_{il}$, where $-1 < \lambda_l < 1$ is the latent coefficient in regressing the lth domain ability on the overall ability $\theta_i \sim N(0,1)$. $\eta_{il} \sim N(0, 1-\lambda_l^2)$ is the error term that is independent of other error terms. Given the overall ability and regression coefficient, $\theta_{il} \mid (\theta_i, \lambda_l) \sim N(\lambda_l\theta_i, 1-\lambda_l^2)$. The correlation between domain abilities θ_{ik} and θ_{il} is $\lambda_k \times \lambda_l$. Note that for this model, an item can only belong to one domain, i.e., the item is simple structured, and the MCMC sampling procedure is different from MIRT in the previous section; it samples the overall ability θ_i from a normal distribution, samples the regression coefficient, and then samples the domain abilities based on the overall ability and the regression coefficients.

The domain abilities are expressed as linear functions of the overall ability, $\theta_{il} = \lambda_l\theta_i + \eta_{il}$. After some notation changes and dropped i, we obtain $\mathbf{Y} = \mathbf{X}\theta + \eta$, where $\mathbf{Y}^T = (\frac{1}{\sqrt{1-\lambda_1^2}}\theta_1, \cdots, \frac{1}{\sqrt{1-\lambda_D^2}}\theta_D)$, $\mathbf{X}^T = (\frac{\lambda_1}{\sqrt{1-\lambda_1^2}}, \cdots, \frac{\lambda_D}{\sqrt{1-\lambda_D^2}})$, $\eta = (\eta_1, \cdots, \eta_D)$, and $\eta_l \sim N(0, \sigma^2)$, for $l = 1, \cdots, D$. σ^2 is close to 1. Suppose the prior $\theta \sim N(0,1)$, then the posterior distribution of the overall ability (Hastie, Tibshirani, & Friedman, 2001) is $\theta \sim N(c\sum_{l=1}^{D}\frac{\lambda_l\theta_l}{1-\lambda_l^2}, c)$, and $c^{-1} = 1 + \sum_{l=1}^{D}\frac{\lambda_l^2}{1-\lambda_l^2}$.

2.6 Multidimensional IRT Rater Models

Suppose there are N examinees, J items. There are M raters with continuous parameter R_r, where $r = 1, \cdots, M$ in the range of $(-\infty, +\infty)$. For a dichotomously-scored item j, with rater R_r, the probability of a correct response to item $j = 1, \cdots, J$ for an examinee with ability $\vec{\theta}_i = (\theta_{i1}, \cdots, \theta_{iD})$, $i = 1, \cdots, N$, for the multidimensional three-parameter logistic (RM-3PL;) model is:

$$P_{ij1r} = P(x_{ijr} = 1 \mid \vec{\theta}_i, \vec{\beta}_j) = \beta_{3j} + \frac{1 - \beta_{3j}}{1 + e^{[-(\vec{\beta}_{2j} \odot \vec{\theta}_i^T - R_r) + \beta_{1j}]}}, \tag{2.9}$$

where $x_{ijr} = 0$ or 1 is the response of examinee i to item j. $\vec{\beta}_{2j} = (\beta_{2j1}, \cdots, \beta_{2jD})$ is a vector of dimension D for item discrimination parameters. β_{1j} is the intercept or the difficulty parameter, β_{3j} is the lower asymptote or the guessing parameter, and $\vec{\beta}_{2j} \odot \vec{\theta}_i^T = \sum_{l=1}^{D} \beta_{2jl}\theta_{il}$. The parameters for the jth item are $\vec{\beta}_j = (\vec{\beta}_{2j}, \beta_{1j}, \beta_{3j})$.

For a polytomously-scored item j, the probability of a response $k-1$ from rater R_r, where $r = 1, \cdots, M$, to item j for an examinee with ability $\vec{\theta}_i$ is given by the multi-dimensional version of the generalized two-parameetr partial credit model (RM-2PPC)

$$P_{ijkr} = P(x_{ijr} = k-1 \mid \vec{\theta}_i, \vec{\beta}_j, R_r) = \frac{e^{(k-1)(\vec{\beta}_{2j} \odot \vec{\theta}_i^T - R_r) - \sum_{t=1}^{k} \beta_{\delta_t j}}}{\sum_{m=1}^{K_j} e^{[(m-1)(\vec{\beta}_{2j} \odot \vec{\theta}_i^T - R_r) - \sum_{t=1}^{m} \beta_{\delta_t j}]}}, \tag{2.10}$$

where $x_{ijr} = 0, \cdots, K_j - 1$ is the response of examinee i to item j. $\beta_{\delta_k j}$ for $k = 1, 2, \ldots, K_j$ are the threshold parameters, $\beta_{\delta_1 j} = 0$, and K_j is the number of response categories for the jth item. The parameters for the jth item are $\vec{\beta}_j = (\vec{\beta}_{2j}, \beta_{\delta_2 j}, \ldots, \beta_{\delta_{K_j} j})$.

For the unidimensional generalized two-parameter partial credit rater model (R-2PPC), we have

$$P_{ij1r} = \frac{1}{\sum_{m=1}^{K_j} e^{[(m-1)(\beta_{2j}\theta_i - R_r) - \sum_{t=1}^{m} \beta_{\delta_t j}]}}, \tag{2.11}$$

$$P_{ij2r} = P_{ij1r} e^{\beta_{2j}\theta_i - R_r - \beta_{\delta_2 j}}, \tag{2.12}$$

$$P_{ijkr} = P_{ij(k-1)r} e^{\beta_{2j}\theta_i - R_r - \beta_{\delta_k j}}, \tag{2.13}$$

$$\log \frac{P_{ijkr}}{P_{ij(k-1)r}} = \beta_{2j}\theta_i - R_r - \beta_{\delta_k j} \tag{2.14}$$

where for $k = 2, \cdots, K_j$, β_{2j} is the discrimination. For a rasch two-parameter partial credit rater model (R-1PC), $\beta_{2j} = 1$. For an ideal rater, $R_r = 0$ and the models are the same as regular multidimensional three-parameter logistic model and generalized two-parameter partial credit model. Normally, multiple choice items are modeled by M-3PL, and there are no rater effect for those items. An CR item with rater scores of two category (0 or 1) can be modeld by M-2PPC, which is the same as M-3PL with guessing=0.

The rater parameters are $\vec{R} = (R_1, \cdots, R_M)$. In MCMC estimation for the rater model, the scale is fixed by fixing the population distributions of standard normal$N(0,1)$, and for each MCMC iteration, the mean of all the raters are 0; or let one of the rater or the first rater has value 0. The prior distribution of the raters is $N(0,1)$. Let

$$P_{ijr} = P_{ijr}(X_{ijr} \mid \vec{\theta}_i, \vec{\beta}_j) = P_{ij1r}^{1_{(X_{ijr}=1)}} (1 - P_{ij1r})^{1_{(X_{ijr}=0)}} \tag{2.15}$$

for RM-3PL item,

$$P_{ijr} = P_{ijr}(X_{ijr} \mid \vec{\theta}_i, \vec{\beta}_j, R_r) = \prod_{k=1}^{K_j} P_{ijkr}^{1_{(X_{ijr}=k-1)}}, \tag{2.16}$$

for RM-2PPC item, and where

$$1_{(X_{ij}=k)} = \begin{Bmatrix} 1 & \text{if } X_{ij} = k \\ 0 & \text{otherwise} \end{Bmatrix}$$

The posterior distribution for the parameters $(\boldsymbol{\theta}, \boldsymbol{\beta}, \vec{R})$ is

$$f(\boldsymbol{\theta}, \boldsymbol{\beta}, \vec{R}) = \prod_{i=1}^{N} \prod_{j=1}^{J} \prod_{r=1}^{M} P_{ijr}(X_{ijr} \mid \vec{\theta}_i, \vec{\beta}_j, R_r) f(\vec{\theta}_i) f(\vec{\beta}_j) f(R_r) \tag{2.17}$$

where $f(\vec{\theta})$ is the density for multivariate normal $N(\vec{0}, \Sigma)$, $f(R_r) \sim N(0,1)$. $f(\vec{\beta}_j) = f(\vec{\beta}_{2j}) f(\beta_{1j}) f(\beta_{3j}) = \prod_{l=1}^{D} f(\beta_{2jl}) f(\beta_{1j}) f(\beta_{3j})$, and $f(\beta_{2jl}) \sim logNormal(0,1)$ for $l = 1, \cdots, D$, $f(\beta_{1j}) \sim N(0,1)$, and $f(\beta_{3j}) \sim beta(6,16)$.

Suppose there are J_1 MC items and J_2 CR items. The response data is arranged as shown in Table 1, with a total of $J_1 + M \times J_2$ columns and N rows.

Table 1

Response Data Layout

Examinee	*MC Items*	CR Items		
		$Rater_1$	$\cdots$	$Rater_M$
1	J_1	J_2	J_2	J_2
2	J_1	J_2	J_2	J_2
...				
N	J_1	J_2	J_2	J_2

If an CR item j for examine i is not scored by rater R_r, then the response for row i and column $J_1+(r-1)\times J_2+j$ is indicated by F.

MCMC can be used to estimate item, ability, and rater-effect parameters.

2.7 NonCompensatory Multidimensional IRT Models

For the noncompensatory model of dimension D, there are D discrimination parameters corresponding to the D difficulty parameters (for M-3PL); similarly for M-2PPC models. The probability Equation 2.1 will become

$$P_{ij1} = P(x_{ij} = 1 \mid \vec{\theta}_i, \vec{\beta}_j) = \beta_{3j} + \frac{1-\beta_{3j}}{\prod_{l=1}^{D} \left(1 + e^{(-\vec{\beta}_{2jl}\vec{\theta}_l + \beta_{1j})}\right)}, \tag{2.18}$$

Chapter 3

Applications for BMIRT

3.1 Working Folders under BMIRTSoftware

After extracting all files from *BMIRTSoftware.zip* using winzip, you should see some working folders, the complied library "lib", and a sample s-plus code for trace plot; their name and features are listed in Table 1. For each working folder, copy "lib" into it and double click the "Files to Run Application", for example "final.bat", you will see a black screen with MCMC iteration. Users need to create "Data File" and "Control File" in the same format as those listed in Table 1 to run their own applications. Detailed explanation of those files in Table 1 follows.

Table 1. Working Folder Name, Model, Bat File to Run Application, Input Data and Control Files

Name	*Models and Function*	*Files to Run Application*	*Feature Commend*	*Data File*	*Control File*
One-Group	M-3PL	final.bat	BMIRT28.bat	G5.rwo	*wt5_1.ctl*
	M-2PPC	final.bat	BMIRT29.bat	G5.rwo	*wt5_2.ctl*
	Fix Item	final.bat	BMIRT30.bat	G5.rwo	*wt5_2.ctl*
		final.bat	BMIRTFixPop.bat	G5.rwo	*wt5_3.ctl*
	HO-IRT	final.bat	BMIRTHO.bat	G5.rwo	*wt5_2F.ctl*
	EAP	final.abt	BMIRTEAP.bat	G5.rwo	*EAPwt5_2F.ctl*
	MAP	final.bat	BayesianModeAbility	G5.rwo	*MAP_1D.ctl*
	MAP	final.bat	BayesianModeAbilityS	G5.rwo	*MAP_2D.ctl*
		finalinf.bat	TestItemInf.bat	G5.rwo	*wt_2inf.ctl*
	Information	overallscore.bat	BMIRTInformation.bat	G5.rwo	*wt_2.ctl*
	OverallScore	overallscore.bat	BMIRTMinSolution	G5.rwo	*wt_2inf.ctl*
	TestResponse	overallscore.bat	BMIRTTRF	G5.rwo	*wt_2TRF.ctl*
MultiGroup	Same Size	final.bat	BMIRT28	all.rwo	*all_1.ctl*, *all_2.ctl*
	Different Size	final.bat	BMIRT1	allV.rwo	$all-1.ctl$, $all-2.ctl$
	M-GR	final.bat	BMIRTGradedResponse	all.rwo	*all_2.ctl*
	NonCompensatory	final.bat	BMIRTNonCompensatory	all.rwo	*all_2.ctl*
	DIF Detection	twoDChi.bat	BMIRTChiSquare	real.rwo	confirm.ctl and chi.ctl
GradedResponse	M-GR	final.bat	BMIRTGradeResponse	G5.rwo	*wt5_2.ctl*
NonCompensatory		driver.bat	BMIRTNonCompensatory	all.rwo	*all_2F.ctl*
FixAnchor		final.bat	BMIRTanchor	ma8.rwo ma8-2F.par ma8-2F.ss	ma8-fixanchor.ctl
Classification		Final1.bat	BMIRTClassification	G5.rwo *wt5_1D.par* *cut1.txt*	*wt5_1D.ctl*
Rater	Rater-effect	final.bat	BMIRTRaschRater	C1.rwo	C1.ctl
		final.bat	BMIRTRater	C1.rwo	C1.ctl
Rasch	Rasch	final.bat	BMIRTRasch	ma8.rwo	*ma8_1.ctl*
Testlet	Testlet-effect	final.bat	BMIRTTestLet	*ma8.rwo*	*ma8_testlet.ctl*

3.2 Input and Output Files for BMIRT

3.2.1 Input Files

lib It is a Library, containing the complied Java program of BMIRT.

.rwo It contains responses of the examinees to the test. The responses can be 0-9, and "F" indicates missing response.

.ctl It is a control file containing the information about the data (number of examinees, number of groups, number of items, number of dimensions, number of iterations, burn-in, parameters for the priors and proposals, item type, and item loadings). The output files from BMIRT are explained below.

3.2.2 Output Files

.param.txt It contains all the MCMC sampling for all the item parameters. Each line presents each iteration.

.theta.txt It contains all the MCMC sampling for all the examinee parameters. Each line presents each iteration after burn-in.

likelihood.txt It contains MCMC results for each iteration with value: -Log(likelihood function), -Log(BayesianLikeliho function).

.par It contains estimates of the final item parameters β.

.ss It contains estimates of the final examinees ability θ.

.dm It contains estimates of population mean μ .

.dv It contains estimates of population variance-covariance matrix σ.

.AIC It contains model fit statistics.

.Ierror It contains MCMC error of item parameters.

.Aerror It contains MCMC error of ability parameters.

.ScoreDistribution It contains Score Distribution for each groups, computed from final item and ability estimates.

.ScoreDis It contains Score Distribution for each groups, computed from MCMC sample of every 50 samples.

.domainscore It contains domain scores. Note this only works for simple structured data currently.

.posterior.mean It contains mean of the population posterior distribution.

.posterior.var It contains variance-covariance of the population posterior distribution.

item.sd It contains the item parameter estimates and variance matrix for MCMC error for the item parameter estimates.

LR.txt It is the output file for Lord Chi-Square test; it contains item number, Lord Chi-Square, and the degree of freedom.

BF.txt It contains model fit statistics: *Group*, *-Liklihood*expconstant1*, *-Posterior Liklihood*expconstant2*, **DIC**, *average of -2log(likelihood)*, *effective number of parameters.*

3.3 File Format

3.3.1 Control File.

.ctl file must be in the following format.

First line has:

(NumberofExaminees pergroup) (numberof items) (numberof groups) (firstLvevl) (middlelevel) (numberof Iteration) (burnin) (numberof Dim) (MaxLevelof 2ppc items) (random seed) (abilityPriorMean) (abilityPriorVar) (abilityPriorCovar) (abilityProposalVar) (abilityProposalCovar) (aPriorMean) (aPriorVariance) (aProposalVariance) (bPriorMean) (bPriorVariance)(bProposalVariance) (cPriorA) (cPriorB)(cProposalDelta) (popMeanVar) (popMeanCor) (popMeanProposalDelta) (popVarVar) (popVarCor) (popVarProposalDelta) (hypAProposalDelta) (hypBProposalDelta); they represents the following:

- NumberofExaminees pergroup: number of examinees in each group/sample size.
- numberof items: Number of total items.
- numberof groups: Number of groups/grades/leves in the data file.
- firstLvevl: Normally, this is 1, indicating the first grade/group.
- middlelevel: Indicating the middle group/grade for multi-group calibration. If only one group, then it is 1.
- numberof Iteration: number of iteration.
- burnin: number of MCMC to be through away.
- numberof Dim: number of Dimensions.
- MaxLevelof 2ppc items: max response catagory number for all the CR items.
- random seed:
- abilityPriorMean : mean of the ability prior distribution.
- abilityPriorVar : Variance of the ability prior distribution.
- abilityPriorCov: Covariance of the ability prior functions.
- abilityProposalVar: Variance of the ability proposal functions.
- abilityProposalCovar: Covariance of the ability proposal functions. Correlations between abilities = abilityProposalVar/abilityProposalCovar.

Note that all the correlations are the same here, if you have more than two-dimension. You will find later on in this document how to vary correlations.

- aPriorMean, aPriorVariance, aProposalVariance: mean, variance, and variance of the prior distributions, and the proposal functions for the discrimination. It is a lognormal function

- bPriorMean, bPriorVariance, bProposalVariance: mean, variance, and variance of the prior distributions, and the proposal functions for the difficulty or threshold. It is a normal distribution.

- cPriorA, cPriorB, cProposalDelta : Guessing parameter c has prior beta(α, β) distribution. They are α, β, and small value for proposal (uniform function). $mean = \frac{\alpha}{\alpha+\beta}$; $variance = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.

  The following parameters should be in the *.ctl* file, but this feature currently is not used.

- popMeanVar, popMeanCor, popMeanProposalDelta: Variance , correlations for the prior of the population mean distribution, and proposal step for the proposal function of the population mean.

- popVarVar, popVarCor, popVarProposalDelta: Variance , correlations for the prior of the population variance distribution, and proposal step for the proposal function of the population variance.

- hypAProposalDelta, hypBProposalDelta: parameters for the proposal of higher level population distributions.

The second line has item type. For example:101134 shows item types for 6 items. 0 means the item is not estimated/used/turnoff; 1 presents multiple choice items with 3PL model; otherwise presents response category for constructed response items. "3" means the response can be 0, 1, 2. For dichotomous scored item (MC) with 2PPC/GR models, which is the same as 3PL model with guessing=0, 2 will be used.

The third line contains information about which item decides the dimension. It must have the same number of items as "numberof Dim". For example "1 5" means dimension=2 and discrimination for the first item is 1 0, discrimination for the 5th item is 0 1, if using *BMIRT30*.

The fourth line to (numberof groups+ 3) line contain the item number for each of the (numberof groups) group.

After (numberof groups+3) line, there are "numberofDim" lines, indicating dimensional loading for each item on each dimension. Each line contains 1 or 0, indicating the item load or unload for that dimension, respectively.

For the last line: There are some differences for using different features.

- Using *BMIRTanchor*, the last line will be similar to second line, except using "1" to indicate that this item (parameter) is fixed, and "0" indicate that this item is to be estimated;
- Using *BMIRTFixPop*, the last line contains population variance-covariance matrix and means.
- Using *BMIRTTestLet*, there will be "numberofDim"-1 lines at the bottom, indicating item number for each testlet.

3.3.2 Response File

.rwo file format: First two lines have item information. Third line has "Group header 01", then followed by responses for each examinee in the first group; followed by "Groupheader 02", then responses for each examinee for the second group, etc. Sample sizes are the same for all groups.

To run a job in each folder, copy *lib* into the folder.

3.4 One-Group for Mixed Models of M-3PL and M-2PPC

In folder One-group, a state writing assessment data has 54 items with 41 MC and 13 CR items (Yao & Schwarz, 2006).

- For MC item, level=1, $\vec{\beta}_j$ is the same as 3PL model in Equation 1 and 2.

- For MC item, level $K_j = 2$, $\vec{\beta}_j$ is as in 3PL model in Equation 1 and 2. Guessing $\beta_{3j} = 0$. This is the same as 2PL model

- For CR item, $\vec{\beta}_j$ is as Equation 3 and 4.

The following analysis can be conducted: If you double click *final.bat*, then some analysis will be conducted.

- "REM": this line will not be executed. So to run a job for a certain line, delete "REM"

- call *BMIRT28* $wt5\_1.ctl$ $G5.rwo$ $out/G5\_1$: this will run one-dimensional calibration.

- for f in (1 2) do call *BMIRT28* $wt5\_f.ctl$ $G5.rwo$ $out/G5\_f$:

  this will run one-dimensional and two-dimensional calibration, respectively, and estimate item and ability parameters. The scale is fixed by fixing the population distribution of standard norm/multinorm.

  BMIRT28Nolikeli is similar to *BMIRT28*, but theree is no ouput for the likelihood function; this feature takes much less memory. Therefore, if you have a really large data matrix and *BMIRT28* will not work, then use *BMIRT28Nolikeli*.

- call $BMIRT29$ $wt5\_2.ctl$ $G5.rwo$ $out/G5\_2$:

  This command will run BMIRT, picking up the last output value as initial value.

- call $BMIRTAbility$ $wt5\_2.ctl$ $G5.rwo$ $out/G5\_2$ $nout/G5\_2$:

  This command will read in item ($G5\_2.par$) and ability ($G5\_2.ss$) files in subdirectory *out*, use $G5\_2.ss$ files as initial values for ability, fix item parameters as in $G5\_2.par$ file, and estimate ability parameters, and out put them in folder *nout*.

- call $BMIRT30$ $wt5\_2.ctl$ $G5.rwo$ $out2/G5\_2$:

  This command will run BMIRT, and the scale is fixed by fixing two item parameters (item 12 and 13, as specified in *.ctl*), with discrimination=(1,0), difficulty=0 for item 12, and discrimination =(0,1), difficulty=0 for item 13. The population distribution, ability parameters and other item parameters will be estimated.

- call $BMIRT31$. For this comand, the rwo files with respopnses of A, B, C, D, E, F indicate respnses of 10, 11, 12, 13, 14, 15. F indicate Missing still. Same meaning for the second line in .ctl file.

- call $BMIRT32$. Reponse A, B, C, D, E,G corresponds to 10,11, 12, 13, 14,15. It works for different sample size for different groups.

- call $BMIRTFixPop$ $wt5\_3.ctl$ $G5.rwo$ $out3/G5\_3$:

  This is a 3-dimensional calibration, with the population distribution to be fixed, with variance-covariance matrix and means indicated at the last line of $wt5\_3.ctl$ file. Output files are in folder *out3*. $BMIRTFixPopNolikeli$ is similar to $BMIRTFixPop$ but has not output for the likelihood function.

- *final1.bat* will be good to run simulation study with many conditions and replications.

- If you want to delete MCMC sampling output file automatically (since they are taking up too much spaces, especially for simulation with many conditions), add del in the *.bat* file, for example, "del $out3/G5\_3-theta.txt$".

3.5 Multi-Group Concurrent Calibration

There are research papers that used multidimensional multi-group features of BMIRT. For example, Yao, L., Patz, R., & Lewis, D., (2003), Patz & Yao (2006, 2007), Lin, P (2008), Kim (2011).

3.5.1 Equal Sample Size

In folder Multi-Group, the data has 5 grades/groups, 191 items in total, with 15 examinees for each group, and common items between adjacent grades. *all_1.ctl*, *all_2.ctl* are for one dimensional and two dimensional calibration, respectively; first level is 1, and middle level is 3; third group population mean and variance are fixed to be standard multi-normal, and all others will be estimated. *call BMIRT28 all_2.ctl all.rwo out/all_2* will run two-dimensional exploratory analysis.

3.5.2 Different Sample Size

In this example, sample sizes are different for different groups of examinees. *call BMIRT1 all-1.ctl all.rwo out/all-1* and *call BMIRT1 all-2.ctl all.rwo out/all-2* will run one-dimensional and two-dimensional analysis, respectively. In *all-1.ctl*, the first few numbers in the first line *5 13 14 15 15 15* indicate the number of groups(5), the sample size for group 1(13), sample size for group 2(14), the sample size for group 3 (15), the sample size for group 4 (15), and the sample size for group 5 (15). All the rest of the .ctl files are the same as runing *BMIRT28*. *BMIRTNolikeli1* is similar to *BMIRT1* but has no likelihood function output.

Use *BMIRTGradeResponse1* for different sample sizes for graded respone model combined with three-parameter logistic model, and the last line of CTL file has the specificed population distribution for the middle grade. Use *BMIRTGradeResponseNolikeli1* for the situation that the data matrix is too large for likelihood function output.

3.5.3 Different Sample Sizes for different groups and the population prior distributions for groups are specified differently

call BMIRTFixPop2 Newall_2.ctl allV.rwo out/all_2 is the command in *temp.bat*. It will run jobs for two-dimensional 5 groups. The priors for the five groups are specified at the last five line in *Newall_2.ctl*.

3.5.4 Computing Lord Chi-Square Test for Detecting DIF

There is a 35 item test for the two groups of examinees. There are 15 items that we know are nonDIF items and the other 20 items will be tested for DIF by running BMIRT using multi-group analysis. The *real.rwo* has the following format:

- Responses for the 15 nonDIF items, followed by 40 columns:
- For group 1 examinees, the first 20 columns are responses, then followed by 20 "F".
- For group 2 examinees, the first 20 columns are "F", followed by 20 responses to the 20 items.

To run DIF, follow the following steps (*twoDchi.bat*):

call *BMIRT28 confirm.ctl real.rwo out/real*:

This will run two-dimensional confirmatory analysis. First 15 items are common between the two groups. output file out/real.item.sd will be used later for DIF analysis.

call *BMIRTChisquare chi.ctl out/real.item.sd out/real*:

The output file real.LR.txt contains the DIF analysis results for the suspected 20 items; it contains *item number*, *chi-square*, and *degree of freedom*.

The $*$.*ctl*(*chi.ctl*) file has $D + 1$ lines with the first line contains: total Item number (55=15+20+20), number of groups(2), number of nonDIF items (15), number of DIF items(20), and number of dimensions (2). Last two lines are dimensional loading information.

The applications can be found at Yao & Li (2010) and Liu, H. Y., Li, C., Zhang, P., & Luo, F. (2012)

NEW FEATURE In section 3.5.3, at the end of the first line of *Newall_2.ctl*, there is a number called "cycle number". This number will specify how many interations are used in computing the file .item.sd——which will be used to compute the Chi-Square Test.

So Lord Chi-Square Test for Detecting DIF can use *BMIRTFixPop2* for different sample sizes with different priors for groups.

3.6 Multidimensional Graded Response Model M-GR

In folder GradedResponse, the samples has 60 items, 1000 examinees. *ma8-1.ctl* and *ma8-2.ctl* are for one dimensional and two dimensional calibration from grade response model. *BMIRTGradeResponseFixPop* is similar to *BMIRTFixPop*, which uses the population distributions specificed in the last line of the CTL file. *BMIRTGradeResponse* is similar to *BMIRT28*.

For output file ending with *GradedR.par*,

- For an MC item, level=1, $\vec{\beta}_j$ is the same as 3PL model in Equation 1 and 2.
- For an MC item, level $K_j = 2$, $\beta_{1j} = -\beta^*_{1j}$, where β^*_{1j} is as in 3PL in Equation 1. $\beta_{3j} = 0$.
- For CR item, the item parameters $\vec{\beta}_j$ were as in Equation 8.

For output file ending with *GradedRRevised.par*,

- For an MC item, level=1, $\beta_{1j} = \frac{\beta^*_{1j}}{||\beta_{2j}||}$, where β^*_{1j} is the same as *GradedR.par*, which is the same as 3PL model in Equation 1 and 2.
- For an MC item, level $K_j = 2$, $\beta_{1j} = -\frac{\beta^*_{1j}}{||\beta_{2j}||} = \frac{\beta^{**}_{1j}}{||\beta_{2j}||}$, where β^*_{1j} is as in *GradedR.par* and β^{**}_{1j} is as in 3PL model in Equation 1. $\beta_{3j} = 0$.
- For CR item, $\beta_{\delta_k j} = -\frac{\beta^*_{\delta_k j}}{||\beta_{2j}||}$, where β^*_{1j} is as in *GradedR.par*. $\beta_{\delta_1 j} \geq \beta_{\delta_2 j} \geq \cdots \geq \beta_{\delta_{K_j-1} j}$

Here $||\beta_{2j}|| = \sqrt{\beta^2_{2j1} + \cdots + \beta^2_{2jD}}$.

Let $\vec{\beta}_j = (\vec{\beta}_{2j}, \beta_{\delta_1 j}, \ldots, \beta_{\delta_{K_j} j})$ be the parameters in *GradedRRevised.par*, then $P^{\star}_{ijk} = 1 - \frac{1}{1+e^{\vec{\beta}_{2j}\odot\vec{\theta} - ||\beta_{2j}||\beta_{\delta_k j}}}$, and when $D = 1$, it is $\frac{1}{1+e^{-\beta_{2j}(\theta - \beta_{\delta_k j})}}$ for $k = 1, \cdots, K_j - 1$.

For multigroup graded response model, use *BMIRTGradeResponse* or *BMIRTGradeResponseFixPop* for equal sample sizes, *BMIRTGradeResponseFixPopNolikeli* for equal sample size with nolikelihood fiunction. For *BMIRTGradeResponseFixPop*, the last line of CTL file has the specificed population distribution for the middle grade. Use *BMIRTGradeResponse1* for different sample sizes, and the last line of CTL file has the specificed population distribution for the middle grade. Use *BMIRTGradeResponseNolikeli1* for the situation that the data matrix is too large for likelihood function output and with different sample sizes for groups.

3.7 Testlet Model

In folder Testlet, there are 60 items and 1000 examinees.

call $BMIRTTestLet$ $ma8\_testlet.ctl$ $ma8\_1.rwo$ $out/ma8$

This runs Testlet model of 6 dimensions of 5 testlet. The bottom five lines contains item numbers for each testlet.

For output file ending with .par is similar to previous defined notation. For file ending with *.testlet*, it contains testlet parameter estimates $\gamma = (\gamma_1, \cdots, \gamma_{D-1})$. The variance of the testlet effect is γ^2.

3.8 Fix Anchor Item Calibration

In folder *FixAnchor*, samples has 60 item, 1000 examinee (Yao & Boughton, 2006).

call $BMIRTanchor$ $ma8\_fixanchor.ctl$ $ma8.rwo$ $ma8\_2F.par$ $ma8\_2F.ss$ $out/ma8-2$.

Last line in $ma8\_fixanchor.ctl$ specify which item to be fixed, using 0 as indicator: item 4, 5, 13, 14, 21, 26, 42, 46, 55 are to be fixed with parameter values as in $ma\_2F.par$. $ma8\_2F.ss$ is some ability estimates that will be read in as starting value.

3.9 Rasch Model

Currently, only work for one-dimensional or multidimensional but with simple structure.

call *BMIRTRasch* $ma8\_1.ctl$ $ma8.rwo$ $out/ma8\_1$: This will estimate item parameters for one-dimensional Rasch model. The ability estimate have some problems. So in order to estimate ability, use other software or call *BMIRTAbility* $ma8\_1.ctl$ $ma8.rwo$ $out/ma8\_1$ $nout/ma8\_1$.

3.10 Rater Models

In folder rater, $final.bat$ will run rater models.

call *BMIRTRaschRater* $C1.ctl$ $C1.rwo$ $out/C1Rasch$:

This will run one-dimensional Rasch rater model. For C1.ctl, everything is the same as runing for BMIRT28 except that the following parameters are added on the first line

- integer value for the number of raters(numRaters),
- number of CR items (numberCR),
- the number for the middle rater that fixed to have value 0 (fixrater),
- double value for the mean distribution of the raters(raterPriorMean),
- double value for the variance of the rater distribution(raterPriorVar),

- double value for the steps in MCMC draw for the rater(raterProposalDelta).

call *BMIRTRaschRaterStartingValue C*1*.ctl C*1*.rwo out/*:

This will continue previous run and the starting values are the estimates from previous run.

call *BMIRTRater C*1*.ctl C*1*.rwo out/C*1:

This will run one-dimesnional rater model(discriminations are estimated).

call *BMIRTRaterStartingValue C*1*.ctl C*1*.rwo out/C*1:

This will continue previous run and the starting values are the estimates from previous run.

call *BMIRTRaschRater C*2*.ctl C*1*.rwo out/C*2*Rasch*:

This will run two-diemsional rasch rater models.

call *BMIRTRater C*2*.ctl C*1*.rwo out/C*2:

This will run two-dimensional rater models.

call *BMIRTGradedResponseRater C*2*.ctl C*1*.rwo out/C*2, will run graded response with rater model.

An application of rasch Rater-effect model can be found at Wang & Yao (2011,2012) and Wei & Yao (2013).

3.11 Computing Overall Score by Domain Scores Using Maximum Information Method

The following step will be used to produce the overall score, and they are in folder One-group with a bat file named *overallscore.bat*.

BMIRTInformation.bat:

BMIRTMinSolution.bat:

call $BMIRTInformation$ $wt5\_2.ctl$ $out/G5\_2.par$ $out/G5\_2.ss$ $out/G5\_2$

Note that the format of $wt5\_2.ctl$ is similar to the ctl file running BMIRT. This step reads in ctl file($wt5\_2.ctl$), item parameter file($out/G5\_2.par$), and D dimensional domain scores($out/G5\_2.ss$), and output file with name $out/G5\_2.Inf$, which contains domain scores, and $D \times D$ information matrix at this domain core point, and the variance matrix (the inversion of the information).

call $BMIRTMinSolution$ $G5\_2inf.ctl$ $out/G5\_2.Inf$ $out/G5\_2$

Note that $G5\_2inf.ctl$ has only one line that contains sample size, dimension, number of iteration(for searching minimum solution),random seed, and steps (in searching minimum solution). For example: 2500 2 4000 923879631 0.01

This step reads in ctl file and information file to produce file named $G5\_2.weighted.score$, which contains D-dimensional domain score, overall score, standard error of measurement for the overall score, and the D dimensional weight.

3.12 Higher-order IRT model for Domain Scores and Overall Scores

In folder One-group, you will find the following for conducting HO-IRT model. Loading has to be simple structure.

call $BMIRTHO$ $wt5\_2F.ctl$ $G5.rwo$ $out/HOG5\_2F$.

This will read input file $wt5\_2F.ctl$ $G5.rwo$ and output file with name $out/HOG5\_2F.overall$, $out/HOG5\_2F.coeff$, $out/HOG5\_2F.ss$, $out/HOG5\_2F.par$, etc.

.overall: contains the overall score.

.coeff: contains coefficient.

.ss: contains domain scores.

.par: contains item parameter estimates.

3.13 NonCompensatory Multidimensional IRT Models

In folder NonCompensatory, *driver.bat* will run a two-dimensional non compensatory model.

3.14 Multidimensional Ability Estimates

3.14.1 Maximum Likelihood estimates (MLE) and Maximum a Posterior Ability Estimates (MAP)

In folder One-group, *.bat* file contains a line:

call BayesianModeAbility MAP_2D*.ctl G5.rwo out*$/G5\_2$*.par out*$/G5\_2$

The commend will read three input files MAP_2D*.ctl G5.rwo out*$/G5\_2$*.par* and output ability estimates in folder *out* with a file named $G5\_2ModeSS$*.txt*.

For MAP_2D*.ctl*, the first line has:

- *number of examinee, number of item, number of dimension, iteration, random seed, step, "1" means MAP or "0" means MLE.*
- Second line presents response levels for all the items.
- Third line presents item number.
- Last line contains variance-covariance matrix and means for the population prior.

3.14.2 Expected a Posterior (EAP)

In folder One-group, *.bat* file contains a line:

call BMIRTEAP $EAPwt5\_2F$*.ctl* $G5.rwoout/G5\_2$*.par* $out/EAPwt5\_2$

The commend will read three input files $EAPwt5\_2F.ctl$ $G5.rwo$ $out/G5\_2.par$ and output EAP estimates in folder out with a file named $G5EAPwt5\_2 - EAP.ss$ (ability estimates in 0/1 metric) and $G5EAPwt5\_2 - EAP - ScaleScore.txt$(Ability estimates in scaled score).

For $EAPwt5\_2F.ctl$, all the lines are the same as running BMIRT28, except the lines after dimensional loading:

upper diagonal element for the population distribution variance-covariance matrix, means for the population distribution.

next g lines (g is the number of groups or grades) contains the lower limit and upper limit for the theta in 0/1 matrix, normally -4 and 4. How many quardrature points, sd and mean of the scaled scores.

A study comparing the performaces of the three methods were conducted in Yao (2013c).

3.15 Computing Test Response Function

In folder One-group, *overallscore.bat* has a line with:

call $BMIRTTRF$ $wt5\_1TRF.ctl$ $out/G5\_1.par$ $out/G5\_1$

This will read in $wt5\_1TRF.ctl$ and $out/G5\_1.par$ and compute the test response function and output a file with name $out/G5\_1.TRF.txt$

3.16 Computing Item and Test Information

In folder One-group, *finalinf.bat* has two lines:

First line will read in $wt5\_1.ctl$ $out/G5\_1.par$ $out/G5\_1.ss$ and output file with name $out/G5\_1.inf$.

Second line will read in $Inf/G5\_1.par$ and output files in folder Inf with informations for each item and test. The first line in $Inf/G5\_1.par$ contains *item number*, *number of dimensions*, *lower theta value for dimension 1* , *higher theta value for dimension 1*, *lower theta value for dimension 2* , *higher theta value for dimension 2,...*, *number of quarture points*, *random seed*.

3.17 MIRT Classification Accuracy and Consistency for both D- and P-method

P-method: classification indices are computed based on the data. D-method: classification indices are computed from multivariate normal distributions $N(\mu, \sigma)$. Here $\mu = 0$ and σ is a variance-covariance matrix, with $var = abilityPriorvar$ and $cov = abilityPriorcovar$ in the ctl file.

CompensatoryClassification.bat will read in ctl file, par file, cut file and output classification results.

call BMIRTClassification $wt5\_1D.ctl$ $wt\_1D.par$ $cut1$ $out/cut1$

$wt5\_1D.ctl$ is similar in running *BMIRT28*.

$cut1.txt$ file contains: number of cut(n), and the n cut scores.

$out/cut1 - wt5\_1D - P.CA.txt$ contains classification accuracy for P-method.

$out/cut1 - wt5\_1D - P.CC.txt$ contains classification consistency for P-method.

$out/cut1 - wt5\_1D - D.CA.txt$ contains classification accuracy for D-method.

$out/cut1 - wt5\_1D - D.CC.txt$ contains classification consistency for D-method.

NonCompensatoryClassification.bat will run noncompensatory model.

3.18 Accessing the Dimensional Structure and Cluster Analysis

Yao & Schwarz (2014) discussed methods for detecting dimensional structure and cluster analysis.

In folder Multi-Group, double click final1.bat, it will run calibration for 2-dimensional model for response data allV.rwo. It has 5 groups and each group has number of examinees of sizes 13 14 15 15 15.

To run cluster analysis with predefined angle of 20 for 2-dimensional calibration, edit the first line of all-2.par to be: 191 2 20 where 191 is the number of items, 2 is the number of dimensions and 20 is the predefined angle. Save all-2.par

Double click *finalfit.bat*, in folder out, you will see files:

all-2.ALR.txt: Contains the output using approximate likelihood ratio test.

all-2.ChiSquareFit.txt: Contains the output for χ_1^2 within dimensions.

all-2.ChiSquareJointFit.txt: Contains the output for χ_2^2 within dimensions.

all-2.Angle.txt: Contains angle between any two pairs of items.

all-2.Cluster.txt: Contains the cluster number and the items in that cluster using the predefined angle.

In folder One-group, you will see examples to do paralle analysis for the number of dimensions.

Double click *ParallAnalysisdriver.bat*—-this will run paralles analysis for data *MA.rwo*. The third line of *MA.rwo* has the number of examinees(1000), number of items (25), number of replications(10), and index for computing method(1 means using pearson correlation, 0 means using tetrachoric correlation). Out put are in folder Engenvalue.

How to Compare? The number of dimensions needed to model the data is the number of eigenvalues that are greater (in the seond line) than those from the random data(the last line).

Chapter 4

Applications for LinkMIRT

4.1 Working Folders under LinkMIRT

After extracting all files from *LinkMIRT.zip* using winzip, you should see some files and the complied library "MEQTlib"; their name and features are listed in Table 2.

Table 2. Bat File to Run Application, features, Input Files, and Output Files

Files to Run Application	*Feature Commend*	method File	*Control File*	*Old item*	*New item*	*Output*
			Linking By Common Item			
driver.bat	MEQUATEGradResponse	SL	Anchor1.ctl	anchor1.par	new.par	*out/new/_SL_G*
driver.bat	MEQUATE	SL	*wt5_2.ctl*	*Awt5_2.par*	*wt5_2.par*	out/wt2
driver.bat	MEQUATEMeanSigma	MS	Anchor1.ctl	anchor1.par	new.par	*out/new_MS*
driver.bat	MEQUATEMeanMean	MM	Anchor1.ctl	anchor1.par	new.par	*out/new_MM*
			Linking By Common Person			
final.bat	MEQUATECommonPerson		base.ss	estimate.ss	item.par	out/Byperson
			Linking By Population			
final.bat	MEQUATECommonPopulation		base.mean, base.var	estimate.mean, estimate.var	estimate.ss, item.par	out/ByPop

Examples using LinkMIRT can be found in Yao (2011), Yao & Boughton (2009), and Yao (2013c).

4.2 Transformation Formulas

Like unidimensional IRT models, the scale for examinees' ability (or item parameters) in the MIRT models has indeterminacy; that is, the parameters are determined up to a linear transformation. The transformation matrix $\mathbf{A}_{D\times D}$ and location vector $\vec{B}_{1\times D}$ can be determined by the following: For an M-3PL item j, let

$$\vec{\beta}^*_{2j} = \vec{\beta}_{2j}\mathbf{A}^{-1}, \tag{4.1}$$

$$\beta^*_{1j} = \beta_{1j} + \vec{\beta}_{2j}\mathbf{A}^{-1}\vec{B}^T, \tag{4.2}$$

$$\beta^*_{3j} = \beta_{3j}. \tag{4.3}$$

For an M-2PPC item, let

$$\beta^*_{\delta_k j} = \beta_{\delta_k j} + \beta_{2j}\mathbf{A}^{-1}\vec{B}^T, \tag{4.4}$$

for $k = 1, \cdots, K_j$. Let $\vec{\theta}^*_i = \vec{\theta}_i\mathbf{A}^T + \vec{B}$,, then the probability of obtaining a certain score on the jth item is not altered, that is $P_{ijk}(\vec{\theta}^*_i, \vec{\beta}^*_j) = P_{ijk}(\vec{\theta}_i, \vec{\beta}_j)$.

$\mathbf{A} = ((\vec{\beta}^*_{2j})^T(\vec{\beta}^*_{2j}))^{-1}(\vec{\beta}^*_{2j})^T\vec{\beta}_{2j}$. $\vec{B} = ((\vec{\beta}^*_{2j})^T(\vec{\beta}^*_{2j}))^{-1}(\vec{\beta}^*_{2j})^T(\beta^*_{1j} - \beta_{1j})$.

To run linkMIRT, download LinkMIRT.zip.

4.3 Linking by Common-item Design

Double click driver.bat, it will run the job and create item parameters after linking. The argument in driver.bat is: *Anchor1.ctl anchor1.par new.par new_ SL*, where

MEQTlib is a Library, containing the complied Java program of LinkMIRT.

Anchor1.par contains the base item parameters, first column is the item number, then by item level(type), discrimination(numDim columns), difficulty, guessing.

New.par contains the item parameters that you want to equate: item number, level discrimination difficulty, guessing.

New_SL is the name you want the item parameter file after linking. The file contains: itemnumber, level, discrimination, difficulty, guessing, 5, numDim *numDim transformation constant A, and numDim location constant B.

4.3.1 Format for the Control File

Format for *Anchor1.ctl* is described as follows: The first line has: numItems, numDim, iterationNumber , delta , ranSeed, numQuarture , low, high, startNumber, deltaM , M1, M2, 1, and they represent the number of items, the number of dimensions, the number of iterations to search, the search steps, Random seed, the number of Quartures on theta, lowerest score on theta, highest score on theta, how many points that start the search, steps for the transformation matrix, initial variance and covariance matrix, initial location parameters, indicator for weight, respectively. If the last number (the indicator for weight) is 1 means the normal density is used as the weight in computing the TRF differences; otherwise it is unweighted. For the second line, it contains item type: 1 presents MC items, 3 means it is a CR item with answer 0, 1, 2. The third line holds the item number that will be used as Equating. The fourth Line has the dimensional loading information this will be used for simple structured test linking.

4.3.2 Storcking-Lord, Meam/Meam, and Mean/Sigma

Please note that:

If the structure is complex, use Multiequatin2PPC, *MEQUATE* is for 3PL+2PPC model using Storking-Lord method.

If the structure is simple, use Multiequatin2PPCSimpleStructure; this can be much faster.

MEQUATEGradedResponse is for M-3P model plus M-GR model using Storking-Lord method.

When the dimension is high, use less quarture points. For example, use 5 quarture points when the dimension is 3.

In .bat file, REM means comment out.

the number of iterations to search can be set high, such as 500-1000. The program will break if the precision level is good enough.

The number of points starting the search can be small, such as 4 or 5.

MEQUATEMeanMean is for mean/mean method.

MEQUATEMeanSigma is for mean/sigma method.

4.4 Linking by Common-person Design

In *final.bat*, the first line is *call MEQUATECommonPerson base.ss estimate.ss item.par out/Byperson*. The base abilities are contained in *base.ss* and the abilities in the new metric are contained in *estimate.ss*. The item parameters in the new metric are contained in *item.par*. After this linking performance, the output files are in *out* with name *Byperson-T.txt*, which has the transformation matrix $\mathbf{A}$, location vector $\vec{B}$, and the abilities in *estimate.ss* converted to the metric of *base.ss*. Another ouput file named *Byperson-T.par* containes the item parameters that were converted to the base scale. Please note that the second line for *base.ss* and *estimate.ss* has the number of examinees, number of common-person, number of dimensions and the number of items.

4.5 Linking by Random Group Design

In *final.bat*, there is a line *call MEQUATECommonPopulation base.mean base.var estimate.mean estimate.var base.ss item.par out/ByPop* . It will convert the abilities and item paramters using the population distributions from the new scale and the base scale.

Chapter 5

Applications for SimuMIRT

5.1 Working Folders under SimuMIRT

After extracting all files from *SimuMIRT.zip* using winzip, you should see some files and the complied library "SimuRwolib"; their name and features are listed in Table 3.

Table 3. Bat File to Run Application, features, Input Files, and Output Files

Files to Run Application	*Feature Commend*	*Item File*	*Output File*
driver.bat	SimulateRwo	test.par	out/test
driver.bat	SimulateGRRwo	test.par	out/GRtest
driver.bat	SimulateRaterRwo	test.rater, test.par	out/ratertest
driver.bat	SimulateNonCompensatoryRwo	test.par	out/NonCompensatorytest
simudriver.bat	SimulateTheta	theta1.ctl	out/S1.theta
simudriver.bat	SimulateSimpleItemParam	ItemPool1.ctl	out/SimpleItemPool1

Download SimuMIRT.zip. *driver.bat* has a line: call *SimulateRwo test.par test* The input file is test.par and the output files are: 1) test.rwo; 2) test-Truetheta.txt (generated theta that were used to generate responses); 3) test-FREQ.txt (containing item parameters and the number of cases for each response). The format for *test.par* is described as follows: The first line has the number of items, the number of examinees, the number of dimensions, population means, population variance-covariance matrix, maximum CR item level, random seed one for generating ability from normal distribution, and Random seed two for generating responses. For simulation with 20 replications, for example, one needs to create 20 par files with different random seed two for generating responses; all others remain the same. Using one bat file such as *(1 2 ··· 20)* (match the name of the par files) will simulate 20 set of responses, but with the same true abilities. The rest of the lines are the item parameters in the format of output *.par* file from BMIRT.

5.2 Responses Following Compensatory MIRT Models

SimulateRwo is for the compensatory MIRT model. **SimulateRwo1** is for the compensatory MIRT model, with known ability. **SimulateGRRwo** is for graded response model. **SimulateGRRwo1** is for the graded response model, with known ability.

5.3 Responses Following NonCompensatory MIRT Models

SimulateNonCompensatoryRwo is for nonCompensatory MIRT model. **SimulateNonCompensatoryRwo1** is for the nonCompensatory MIRT model with known ability.

5.4 Responses Following Rater Effect Models

SimulateRaterRwo is for the compensatory MIRT model with rater effect. **SimulateRaterRwo1** is for the compensatory MIRT model with rater effect and knowm ability.

5.5 Simulate Abilities and Item Parameters

simudriver.bat has three lines. The first line will simulate item parameters, the second line will simulate item parameters of simple structured, and the last line will simulate abilities. The input files are explained below:

For *ItemPool2.ctl*, the first line has the number of items, dimension, random seed, mean and variance for the discrimination parameetrs, mean and variance for the difficulty parameters, and beta parameters for the guessing. For the second line, it has all the the item types.

For *ItemPool1.ctl*, it contains the number of total items, the number of dimension, the number of items for each dimension, random seed, mean and variance for the discrimination parameters, mean and variance for the difficulty parameters, and beta parameters for the guessing, lower and upper limit for the discrimination, lower and upper limit for the difficulty.

For *Theta.ctl*, it contains the number of simulees, the dimension, the means and variance-covariance matrix for the population distributions, and random seed.

The ouput files are the item parameters and the abilities.

Chapter 6

Multidimensional Computer Adaptative Test

Five multidimensional computer adaptive testing (MCAT) item selection procedures are developed with two methods for the item exposure control and the Priority Index (PI) method for the content constraints. One item exposure control method is the Sympson-Hetter procedure (SH, 1985) and the other is to simply put a limit on the item exposure rate ($probability$), all in the MCAT frame work. The five procedures are: Volume (Vm, Segall, 1996), Kullback-Leibler information (KL, Veldkamp & van der Linden, 2002), Minimize the error variance of the linear combination (V_1, van der Linden, 1999), Minimum Angle (Ag, Reckase, 2009), and Minimize the error variance of the composite score with the optimized weight (V_2, Yao, 2010). For each of the five procedures, there are different procedures regarding content constraints and item exposure rate. User needs to specify true examinees, item pools and procedures. The output file will have the estimated ability and selected items for each simulee.

6.1 Working Folders under SimuMCAT

After extracting all files from *SimuMCAT.zip* using winzip, you should see files and the complied library "lib". Their name and features are listed in Table 4 for "Angle" method; other item selection methods are named in the same fashion.

Table 4. Bat File to Run Application, Features, Input Files, and Output Files

Files to Run Application	*Feature Commend*	*Input File*	*OutPut File*
final.bat	CATItemSelectionByAngle	CTL/S1000-4D-C1-1-10itemTruetheta.txt	ByAngle/c1/S1000-4D-C1-1-10.ss
	Angle	ItemPool/AFQT.par	ByAngle/c1/S1000-4D-C1-1-10.par
final.bat	CATItemSelectionByAngleContent	CTL/S1000-4D-C1-1-10itemTruetheta.txt	ByAngle/c2/S1000-4D-C1-1-10.ss
	Angle with Content Control	ItemPool/AFQT.par	ByAngle/c2/S1000-4D-C1-1-10.par
final.bat	CATItemSelectionByAngleContentOrder	CTL/S1000-4D-C1-1-10itemTruetheta.txt	ByAngle/c3/S1000-4D-C1-1-10.ss
	Angle with content alternatining	ItemPool/AFQT.par	ByAngle/c3/S1000-4D-C1-1-10.par
final.bat	CATItemSelectionByAnglePriorityIndex	CTL/S1000-4D-C1-1-10itemTruetheta.txt	ByAngle/c4/S1000-4D-C1-1-10item.ss
	Angle with priority index	ItemPool/AFQT.par	ByAngle/c4/S1000-4D-C1-1-10item.par
final.bat	CATItemSelectionByAnglePrecision	CTL/S1000-4D-C1-1-10itemTruetheta.txt	ByAngle/c6/S1000-4D-C1-1-10item.ss
	Varibale-length SE as stopping rule	ItemPool/AFQT.par	ByAngle/c6/S1000-4D-C1-1-10item.par
final.bat	CATItemSelectionByAngleSE	CTL/S1000-4D-C1-1-10itemTruetheta.txt	ByAngle/c7/S1000-4D-C1-1-10item.ss
	Varibale-length PSER as stopping rule	ItemPool/AFQT.par	ByAngle/c7/S1000-4D-C1-1-10item.par

6.2 Multidimensional CAT Item Selection Methods

6.2.1 Kullback-Leibler Information (KL)

For a M-3PL item m, the Kullback–Leibler information is the distance between two likelihoods at two ability points $\vec{\theta}^{j-1} = (\theta_1^{j-1}, \cdots, \theta_D^{j-1})$ and $\vec{\theta}_0$ and is defined as:

$$K_m(\vec{\theta}^{j-1}, \vec{\theta}_0) = E_{\vec{\theta}_0} \log[\frac{P_m(X_m \mid \vec{\theta}_0, \vec{\beta}_m)}{P_m(X_m \mid \vec{\theta}^{j-1}, \vec{\beta}_m)}] = P_{m1}(\vec{\theta}_0) \log \frac{P_{m1}(\vec{\theta}_0)}{P_{m1}(\vec{\theta}^{j-1})} + (1 - P_{m1}(\vec{\theta}_0)) \log \frac{1 - P_{m1}(\vec{\theta}_0)}{1 - P_{m1}(\vec{\theta}^{j-1})}, \quad (6.1)$$

where $\vec{\theta}_0$ is the true ability, and $\vec{\theta}^{j-1}$ is the current ability estimates based on selected $j-1$ items. The Kullback–Leibler information tells us how well the response variable discriminates between the ability estimates and the true ability value. For $j-1$ selected items, define $\mathbf{K_{j-1}}(\vec{\theta}^{j-1}, \vec{\theta}_0) = \sum_{l=1}^{j-1} K_l(\vec{\theta}^{j-1}, \vec{\theta}_0)$. The Bayesian KL for item m

(Chang & Ying, 1996; Veldkamp & van der Linden, 2002) is

$$K_m(\vec{\theta}^{j-1} \mid \vec{X}) = \int_{\vec{\theta}} (\mathbf{K}_{j-1}(\vec{\theta}^{j-1}, \vec{\theta}) + K_m(\vec{\theta}^{j-1}, \vec{\theta})) f(\vec{\theta} \mid \vec{X}) d\vec{\theta}$$

$$= \int_{\theta_1^{j-1}-\delta_j}^{\theta_1^{j-1}+\delta_j} \cdots \int_{\theta_D^{j-1}-\delta_j}^{\theta_D^{j-1}+\delta_j} (\mathbf{K}_{j-1}(\vec{\theta}^{j-1}, \vec{\theta}) + K_m(\vec{\theta}^{j-1}, \vec{\theta})) f(\vec{\theta} \mid \vec{X}) d\theta_1 \cdots \theta_D \tag{6.2}$$

where $\delta_j = \frac{3}{\sqrt{j}}$.

1. For each item m in the pool, compute the posterior KL information $K_m(\vec{\theta}^{j-1} \mid \vec{X})$ using Equation 12. Here $\vec{X}$ is the response vector for the selected $j-1$ items.

2. Select item $j = m$ such that $K_m(\vec{\theta}^{j-1} \mid \vec{X})$ has the maximum value.

3. Update ability $\vec{\theta}^j$ based on the selected j items.

6.2.2 Volume (Vm)

In Segall (1996), he proposed selecting the next item j by maximizing the determinant of the posterior information as follows:

$$W = |\mathbf{I}_{j-1}(\vec{\theta}^{j-1}) + I_j(\vec{\theta}^{j-1}) + \Sigma^{-1}|, \tag{6.3}$$

where $\mathbf{I}_{j-1}(\vec{\theta}^{j-1})$ is the information obtained from already selected $j-1$ items at the ability estimates $\theta^{\vec{j-1}}$.

1. For each item m in the pool, compute the volume or the determinant of the information using

$$W_m = |\mathbf{I}_{j-1}(\vec{\theta}^{j-1}) + \frac{(P_{m1} - \beta_{3m})^2 (1 - P_{m1})}{P_{m1}(1 - \beta_{3m})^2} \vec{\beta}_{2m} \otimes \vec{\beta}_{2m} + \Sigma^{-1}|$$

   at ability $\vec{\theta}^{j-1}$.

2. Select item $j = m$ such that W_m has the maximum value.

3. Update ability $\vec{\theta}^j$ and information $I_j(\vec{\theta}^j)$ based on the selected j items.

For the non-Bayesian procedure, the above equations still hold with the removal of Σ^{-1}. However, the first D items must be selected from the D domains, especially for the items of simple structure; as the matrix needs to be non-singular.

6.2.3 Minimize the Error Variance of the Composite Score with the Optimized Eeight (V_2)

For a test with J items of known item parameters, for a given score point $\vec{\theta}$, the test information is $\mathbf{I}_J(\vec{\theta})$. the composite score $\theta_{\vec{\alpha}} = \sum_{l=1}^{D} \theta_l w_l$ has a standard error of measurement $SEM(\theta_{\vec{\alpha}}) = V(\theta_{\vec{\alpha}})^{1/2}$, where $V(\theta_{\vec{\alpha}}) = \vec{w}V(\vec{\theta})\vec{w}^T$, $\vec{w} = (w_1, \cdots, w_D) = (cos^2\alpha_1, \cdots, cos^2\alpha_D)$. $V(\vec{\theta})$ can be approximated by $I(\vec{\theta})^{-1}$. The weight $\vec{w}$, called optimized weight, such that $SEM(\theta_{\vec{\alpha}})$ has a minimum value does exist (Yao, 2011). The weight for selecting j items $\vec{w_j} = \vec{w}$ is the optimized weight derived on the estimated domain abilities and the elected items.

The following steps are used in selecting items for V_2. Let $M < J$ be a chosen integer.

1. For $j <= M$, the weight is pre-fixed weight of equal values, i.e., $\vec{w}_{j-1} = (w_1, \cdots, w_D), w_l = 1/D$ for $l = 1, \cdots, D$.

2. For $j > M$, compute the optimized weight $\vec{w}_{j-1}$ based on the $j-1$ selected items.

3. Select item $j = m$ such that $\vec{w}_{j-1}[\mathbf{I}_{j-1}^m(\vec{\theta}^{j-1})]^{-1}(\vec{w}_{j-1})^T$ has a minimum value.

4. Update ability $\vec{\theta}^j$ and information $\mathbf{I}_j(\vec{\theta}^j)$ based on the selected j items.

The integer M can be chosen by the user. For example, $M = 0$ or $M = \frac{J}{3}$, where J is the total number of selected items. $M = 0$ is applied in this study; pre-run shows that results from $M = 0$ and $M = \frac{J}{3}$ are similar.

6.2.4 Minimize the Error Variance of the Linear Combination (V_1)

This method was studied in van der Linden (1999) for increasing the precision for overall scores. It is similar to V_2, with the weight $\vec{w}_{j-1}$ being pre-fixed with equal values for all $j = 1, \cdots, J$, i.e., $\vec{w}_{j-1} = (w_1, \cdots, w_D), w_l = 1/D$ for $l = 1, \cdots, D$.

6.2.5 Minimum Angle (Ag)

1. At the ability level $\vec{\theta}^{j-1}$, let the direction $\vec{\alpha} = (\alpha_1, \cdots, \alpha_D)$ be the minimizer such that $\cos(\vec{\alpha})\mathbf{I}_{j-1}(\vec{\theta}^{j-1})\cos(\vec{\alpha})^T$ has a minimum value for all possible angles. Here $\cos(\vec{\alpha}) = (\cos\alpha_1, \cdots, \cos\alpha_D)$.

2. For each item m in the pool, compute

$$\mathbf{I}_j^m(\vec{\theta}^{j-1}) = \mathbf{I}_{j-1}(\vec{\theta}^{j-1}) + \frac{(P_{m1} - \beta_{3m})^2(1 - P_{m1})}{P_{m1}(1 - \beta_{3m})^2}\vec{\beta}_{2m} \otimes \vec{\beta}_{2m}$$

   at ability $\vec{\theta}^{j-1}$.

3. Select item $j = m$ such that $\cos(\vec{\alpha})\mathbf{I}_j^m(\vec{\theta}^{j-1})\cos(\vec{\alpha})^T$ has a maximum value (among all the items in the pool).

4. Update ability $\vec{\theta}^j$ and information $\mathbf{I}_j(\vec{\theta}^j)$ based on the selected j items.

6.3 Stopping Rules

A CAT selection process is a cyclical procedure that is stopped by a stopping rule (Reckase, 2009; Wainer, 2000). The stopping rule can be when a specified number of test items has been administered (fixed-length), when the estimated ability has reached the desired precision level, or when a decision has been made with the desired confidence level (varying length).

6.3.1 Fixed-length CAT

The test length is fixed and specified by the user in the CTL file.

6.3.2 Varying Length CAT–the Standard Error (SE) and the Predicted Standard Error Reduction (PSER) Stopping Rules

Let $\vec{P} = (p_1, \cdots, p_D)$ represents the required SEM for the D domain ability estimates; the smaller the SEM, the larger the precision. Let $\hat{P} = (\hat{p}_1, \cdots, \hat{p}_D)$ be the SEM estimates based on the current selected items. If for some domain l, the precision has been achieved, then the items loading in domain l will not be selected anymore. If an item has been selected more times and has reached the required exposure rate, then it will be not be selected anymore. If the number of selected items has reached the maximum limit for certain domain, then no more items will be selected from that domain. At the beginning of a selection process, $p_l < \hat{p}_l$, the selection process stops for that domain if $p_l \geq \hat{p}_l$. For the SE stopping rule, there are some problems. For some examinees, the administered test is lengthy, with too many items being administered without an accompanying improvement in precision. Therefore, a modified procedure (PSER) that predicts the reduction of the SE is proposed and compared with the SE method. For PSER, there are two modifications: 1) a predetermined parameter α is applied, and if the SEM reduction based on the current selected items and the previously selected items is smaller than α, then the item selection for this domain is stopped, even if the SE requirement has not been met; 2) a predetermined parameter β is applied, and if the SE reduction based on the current selected items and the previously selected items is larger than β, then the item selection for this domain will continue with a slightly larger weight (adding .0001), even if the SE requirement has been met. For unidimensional IRT, the information is a monotonically increasing function with respect to the number of items administered. However, this is not the case with the MIRT models. At each score point vector, the information is a matrix and the directional information along each of the dimensions/domains is not a monotonic function with respect to the number of items administered. Therefore, extra rules are applied. They are: 3) if the current precision is not smaller than the previous step and the current precision is within α distance away from the required precision, then stop selecting items from this domain; 4) if the current precision (SEM) is not smaller than the previous step and the current precision is outside $2 \times \beta$ distance away from the required precision, then the item selection for this domain will continue with slightly higher weight (adding .0001). Please note that α and β can be specified differently and $\beta > \alpha$. β measures how much SEM reduction you would allow to select one more item to increase the precision. α measures how much SEM reduction that you can tolerate to keep selecting items. The rules for PSER ensure that (a) lengthy tests are prevented when the pool has no more quality items that will improve the precision of the examinee's ability estimates; (b) better precision is obtained with one or two more items; and (c) the precision is not much worse than the required precision.

Please note those SEMs or the precisions for the domain abilities and the overall ability can be implemented along with the Priority Index; the choice depends on the purpose of the test.

Chapter 7

Application of SimuMCAT

7.1 Input Files

Input files for SimuMCAT are listed below, representing %1 %2 %3:

lib: Library, containing the complied Java program of SimuMCAT.

%1, .txt: Contains the true abilities for the simulated examinees.

%2, .par: It contains the item parameters in the item pool. The order is: the vector of discriminations, difficulty or threshold, guessing, objective for this item. At the bottom, there are *numDim* lines indicating the loading information for all the items.

%3, .ss, .examinee.par: The ouput files .ss and .par has the estimated abilities and selected items, respectively.

Explanations for *.txt* file

The first line.

- numAllitems: integer value, representing the total number of items in the pool.
- numExaminee: integer value, representing the number of simulees.
- numDim: integer value, indicating the number of dimensions.
- numObj: integer value, indicating the number of objectives.
- SelectedItem: integer value, indicating the number of selected items.
- there are $2 \times numObj$ integer values representing the following: lower limit for objective 1, upper limit for objective 1, etc.
- numIterations: integer value indicating the number of iterations for MAP ability estimates.
- numIterationsAngle: integer value presenting the number of iterations in searching the minimum angle.
- ranSeed: integer values for random seeds in generating ability and other use for random needs.
- Delta: Double and small value.
- DeltaAngle Double and small value used in searching for minimum angle.
- Bayesian: integer value with " 1" indicating Bayesian, "0" indicating NonBayesian, "2" indicating that the first few items are selected by Bayesian and later selections are by NonBayesian.
- rate: Double value representing the limit or the maximinum for item exposure rate, for example " 0.3".
- two double values indicate the weight for the precision requirement and the maximinum item number requirement, respectively.
- double vector for the precision requirement for the domains.

- integer top1 indicate that the first item is selected randomly from the top1 ordered items.

- integer top2 indicate that the second item is selected randomly from the top2-3 ordered items, the third item is selected randomly from the top2-6 ordered items.. when the selected item number is smaller than the the number of objectives.

- double value for α which measures how much SEM reduction that you can tolerate to keep selecting items.

- double value for β which measures how much SEM reduction you would allow to select one more item to increase the precision. $\alpha < \beta$.

The second line.

Second line contains initial angle (alpha), indicator for fixing the angle or not (fixangle), indicator of computing test reliability(ReliabilityIndex), and indicator of simple structure or complex structure(StructureIndex).

- alpha: *numDim* double values, indicating the initial angle, for example "1 0 0 0" for 4 domains.

- fixangle: Integer value with "0" means not fix (for example, *Ag* method), "1" means fix angle (for example, for V_1 procedure using simple average). "2" means V_2 use the given weight for the overll score for the first 1/3 of item selections, after that, use the optimized weight. if fixangle=0, then the final overall score is derived based on the optimized weight of the final domain scores; otherwise, the final overall score is the linear combination of the domain scores based on the given weight.

- ReliabilityIndex: "1" compute test reliability (taking longer), "0" means not to compute test reliability;

- StructureIndex: "1" means simple structure, "0" means items are complex structured.

The third line.

Third line contains prior of the population

- Upper triangle for the population variance-covariance matrix.

- Population means.

The rest of the lines contains all the true abilities and the weight for the overall ability.

7.2 Output Files

The output files from SimuMCAT are explained below.

- *.ss*: Contains the examinees ability information. The first line tells you the layout of the file. "True Ability", "True Overall ability", "Estimated Ability", "Estimated Overall Ability", and "their SES(domain and overall)", "time used (unit=second)", "test reliability"

- *.par*: Contains the selected items, responses, ability estimates and their standard errors of measurement, for each examine.

7.3 Content Constraints

7.3.1 No Content Constraints

The procedures for Vm, Ag, KL, V_1, and V_2 without any content constraints are listed below.

call CATItemSelectionByDet %1 %2 %3.: This will select items using Vm method.

call CATItemSelectionByAngle %1 %2 %3.: This will select items using Ag method.

call CATItemSelectionKullbackInf %1 %2 %3.: This will select items using KL method.

call CATItemSelectionByVariance %1 %2 %3.: This will select items using V_2 method.

7.3.2 Fixed Number of Items with Order

The procedures for Vm, Ag, KL, V_1, and V_2 for each content has to have the required number of items and the order alternating among contents are listed below.

call CATItemSelectionByDetContentOrder %1 %2 %3.: This will select items using Vm method.

call CATItemSelectionByAngleContentOrder %1 %2 %3.: This will select items using Ag method.

call CATItemSelectionKullbackInfContentOrder %1 %2 %3.: This will select items using KL method.

call CATItemSelectionByVarianceContentOrder %1 %2 %3.: This will select items using V_2 method.

7.3.3 Fixed Number of Items without Order

The procedures for Vm, Ag, KL, V_1, and V_2 with each content has to have the required number of items are listed below.

call CATItemSelectionByDetContent %1 %2 %3.: This will select items using Vm method.

call CATItemSelectionByAngleContent %1 %2 %3.: This will select items using Ag method.

call CATItemSelectionKullbackInfContent %1 %2 %3.: This will select items using KL method.

call CATItemSelectionByVarianceContent %1 %2 %3.: This will select items using V_2 method.

7.4 Exposure Control and Priority Index

7.4.1 Sympson-Hetter Procedure with Priority Index

%1 %2 %3 %4 presents *.txt* (true abilities), *.par* (item parameters), *.exposure* (Exposure rate table), and out put files. The *.exposure* is a file contains the exposure table obtained from SH method and it will be explained later in this section.

call CATItemSelectionByDetPriorityIndexExposure %1 %2 %3 %4.: This will select items using Vm method.

call CATItemSelectionByAnglePriorityIndexExposure %1 %2 %3 %4.: This will select items using Ag method.

call CATItemSelectionKullbackInfPriorityIndexExposure %1 %2 %3 %4.: This will select items using KL method.

call CATItemSelectionByVariancePriorityIndexExposure %1 %2 %3 %4.: This will select items using V_1 method.

call CATItemSelectionByVariancePriorityIndexExposure1 %1 %2 %3 %4.: This will select items using V_2 method.

The *.exposure* file is obtained below: *call CATTraining.InfoTable ItemPool/AFQT2.par ByVolume/AFQT* reads in the item parameter file and output information table by Volume method. The first line of the item parameter file *ItemPool/AFQT2.par* contains :

numAllitems: integer value indicates item numbers in the pool.

SelectedItem: integer value indicates the number of selected items.

numDim: integer value indicate number of dimensions.

ranSeed : integer value.

quardrature: integer value indicate the number of quardrature ponits in creating the information tables.

low: double value indicate the lower theta values.

high: double value indicate the upper theta values.

replication: integer value indicate number of replications in S-H procedure.

Bayesian: integer, 1–Bayesian method, 0–nonBayesian method, 2–mix, first few items use bayesian and the rest use nonBayesian.

Method : integer, 1 indicate Angle method, 2 indicate volume method, 3 indicate variance method, 4 indicate KL method.

M1 : double for the upper trangle of the matrix for the prior distributuion.

M2 : double vectore for the mean for the prior distributuion.

r : double for the maximum item exposure rate.

$\vec{\alpha}$: double vector indicate initial angle.

fixangle: integer value with 1 means fix angle, 0 means not fix.

numIterations: integer value indicates the number of iterations in searching for angle.

The output file *ByVolume/AFQTinfotable.txt* contains the information table by the theta values. *call CATTraining.SHExposure* ItemPool/AFQT2.par *ByVolume/AFQTinfotable.txt ByVolume/AFQTexposure.txt* will read in *ItemPool/AFQT2.par* and *ByVolume/AFQTinfotable.txt* and produce the exposure tables at *ByVolume/AFQTexposure.txt.*

The information tables are created based on different methods. Use *ItemPool/AFQT1.par*, then the information table is created based on Angle method. Use *ItemPool/AFQT4.par* then the information table is created based on KL method. Use *ItemPool/AFQT3.par* then the information table is created based on variance method.

7.4.2 Probability with Priority Index

The procedures for *Vm*, *Ag*, *KL*, V_1, and V_2 using Priority Index and probability for item exposure control are listed below.

call CATItemSelectionByDetPriorityIndex %1 %2 %3.: This will select items using Vm method.

call CATItemSelectionByAnglePriorityIndex %1 %2 %3.: This will select items using Ag method.

call CATItemSelectionKullbackInfPriorityIndex %1 %2 %3.: This will select items using KL method.

call CATItemSelectionByVariancePriorityIndex %1 %2 %3.: This will select items using V_1 method.

call CATItemSelectionByVariancePriorityIndex1 %1 %2 %3.: This will select items using V_2 method.

7.5 SE Stopping Rules

call CATItemSelectionByKullbackInfPrecision %1 %2 %3.: This will select items using KL method.

call CATItemSelectionByDetPrecision %1 %2 %3.: This will select items using Vm method.

call CATItemSelectionByVariancePrecision1 %1 %2 %3.: This will select items using V_2 method.

7.6 PSER Stopping Rules

call CATItemSelectionByKullbackInfSE %1 %2 %3.: This will select items using KL method.

call CATItemSelectionByAngleSE %1 %2 %3.: This will select items using Ag method.

call CATItemSelectionByDetSE %1 %2 %3.: This will select items using Vm method.

call CATItemSelectionByVarianceSE %1 %2 %3.: This will select items using V_2 method.

Chapter 8

Appendix

8.1 Convergence Issue for MCMC

You may adjust parameters for the prior and proposals, to aim for accept rate between 20-40 percent.

You can run a few chains by changing random seeds and obtain the final estimation by averaging over the few chains.

You can run some iteration, and use the obtained item and ability estimates as the starting value or update the parameters for the priors and proposals based on the estimates and continue to run more iterations.

Write some code (R or S) to check trace plot, eg, S-plus code CheckStationary.ssc in the package.

Other available package to check stationary, eg, BOA at http://www.public-health.uiowa.edu/boa/

8.2 MCMC Algorithms

In BMIRT, the estimation of parameters (θ, β, λ) in the model are obtained by MCMC sampling from the posterior distribution $P(\theta, \beta, \lambda \mid X, Y, Z)$. The sampling procedures are as follows:

8.2.1 Steps to Sample Item Parameters

Sample each β_j^m, $j = 1, 2, \cdots, J$ from $P(\beta_j \mid \beta_{<j}^m, \beta_{>j}^{m-1}, \theta^m, \lambda^{m-1}, X, Z_j, Y)$ as follows:

- Draw $\beta_j^* \sim q_m(\beta_j \mid \beta_j^{m-1})$.

- Calculate the vector of J acceptance probabilities

$$\alpha_j^* = min\{\frac{P(\beta_j^* \mid \beta_{<j}^m, \beta_{>j}^{m-1}, \theta^m, \lambda^{m-1}, X, Z_j, Y) q_m(\beta_j^{m-1} \mid \beta_j^*)}{P(\beta^{m-1} \mid \beta_{<j}^m, \beta_{>j}^{m-1}, \theta^m, \lambda^{m-1}, X, Z_j, Y) q_m(\beta_j^* \mid \beta_j^{m-1})}, 1\}, \tag{8.1}$$

for $j = 1, 2 \cdots, J$, where

$$\frac{P(\beta_j^* \mid \beta_{<j}^m, \beta_{>j}^{m-1}, \theta^m, \lambda^{m-1}, X, Y, Z_j) q_m(\beta_j^{m-1} \mid \beta_j^*)}{P(\beta^{m-1} \mid \beta_{<j}^m, \beta_{>j}^{m-1}, \theta^m, \lambda^{m-1}, X, Y, Z_j) q_m(\beta_j^* \mid \beta_j^{m-1})}$$

$$= \frac{P(X \mid \beta_{<j}^m, \beta_j^*, \beta_{>j}^{m-1}, \theta^m, Z_j) P(\theta^m \mid \lambda^{m-1}, Y) P(\beta_{<j}^m, \beta_j^*, \beta_{>j}^{m-1} \mid Z_j) P(\lambda^{m-1} \mid Y) q_m(\beta_j^{m-1} \mid \beta_j^*)}{P(X \mid \beta_{<j}^m, \beta_j^{m-1}, \beta_{>j}^{m-1}, \theta^m, Z_j) P(\theta^m \mid \lambda^{m-1}, Y) P(\beta_{<j}^m, \beta_j^{m-1}, \beta_{>j}^{m-1} \mid Z_j) P(\lambda^{m-1} \mid Y) q_m(\beta_j^* \mid \beta_j^{m-1})}$$

$$= \frac{\prod_{i=1}^N P_{i,j}(X_{i,j} \mid \theta_i^m, \beta_j^*, Z_j) P(\beta_j^* \mid Z_j) q_m(\beta_j^{m-1} \mid \beta_j^*)}{\prod_{i=1}^N P_{i,j}(X_{i,j} \mid \theta_i^m, \beta_j^{m-1} Z_j) P(\beta_j^{m-1} \mid Z_j) q_m(\beta_j^* \mid \beta_j^{m-1})}. \tag{8.2}$$

- Accept each $\beta_j^m = \beta_j^*$ with probability α_j^*; otherwise let $\beta_j^m = \beta_j^{m-1}$.

8.2.2 Steps to Sample Proficiency:

Sample each θ_i^m, $i = 1, 2 \cdots, N$ from $P(\theta_i \mid \theta_{<i}^m, \theta_{>i}^{m-1}, \beta^{m-1}, \lambda^{m-1}, X, Y_i, Z)$ as follows:

- Draw $\theta_i^* \sim q_m(\theta_i \mid \theta_i^{m-1})$ independently for each $i = 1, 2 \cdots, N$.

- Calculate the vector of N acceptance probabilities:

$$\alpha_i^* = min\{\frac{P(\theta_i^* \mid \theta_{<i}^m, \theta_{>i}^{m-1}, \beta^{m-1}, \lambda^{m-1}, X, Y_i, Z) q_m(\theta_i^{m-1} \mid \theta_i^*)}{P(\theta^{m-1} \mid \theta_{<i}^m, \theta_{>i}^{m-1}, \beta^{m-1}, \lambda^{m-1}, X, Y_i, Z) q_m(\theta_i^* \mid \theta_i^{m-1})}, 1\}, \tag{8.3}$$

for $i = 1, 2 \cdots, N$. where

$$\frac{P(\theta_i^* \mid \theta_{<i}^m, \theta_{>i}^{m-1}, \beta^{m-1}, \lambda^{m-1}, X, Y_i, Z) q_m(\theta_i^{m-1} \mid \theta_i^*)}{P(\theta^{m-1} \mid \theta_{<i}^m, \theta_{>i}^{m-1}, \beta^{m-1}, \lambda^{m-1}, X, Y_i, Z) q_m(\theta_i^* \mid \theta_i^{m-1})}$$

$$= \frac{P(X \mid \theta_i^*, \theta_{<i}^m, \theta_{i>}^{m-1}, \beta^{m-1}, Z) P(\theta_i^*, \theta_{<i}^m, \theta_{i>}^{m-1} \mid \lambda^{m-1}, Y_i) P(\beta^{m-1} \mid Z) P(\lambda^{m-1} \mid Y_i) q_m(\theta_i^{m-1} \mid \theta_i^*)}{P(X \mid \theta_i^{m-1}, \theta_{<i}^m, \theta_{i>}^{m-1}, \beta^{m-1}, Z) P(\theta_i^{m-1}, \theta_{<i}^m, \theta_{i>}^{m-1} \mid \lambda^{m-1}) P(\beta^{m-1} \mid Z) P(\lambda^{m-1} \mid Y_i) q_m(\theta_i^* \mid \theta_i^{m-1})}$$

$$= \frac{\prod_{j=1}^J P_{i,j}(X_{i,j} \mid \theta_i^*, \beta_j^{m-1}, Z_j) P(\theta_i^* \mid \lambda^{m-1}, Y_i) q_m(\theta_i^{m-1} \mid \theta_i^*)}{\prod_{j=1}^J P_{i,j}(X_{i,j} \mid \theta_i^{m-1}, \beta_j^{m-1}, Z_j) P(\theta_i^{m-1} \mid \lambda^{m-1}, Y_i) q_m(\theta_i^* \mid \theta_i^{m-1})} \tag{8.4}$$

$$= \frac{\prod_{j=1}^J P_{i,j}(X_{i,j} \mid \theta_i^*, \beta_j^{m-1}, Z_j) P(\theta_i^* \mid \mu_g^{m-1}, \sigma_g^{m-1}, Y_i = g) q_m(\theta_i^{m-1} \mid \theta_i^*)}{\prod_{j=1}^J P_{i,j}(X_{i,j} \mid \theta_i^{m-1}, \beta_j^{m-1}, Z_j) P(\theta_i^{m-1} \mid \mu_g^{m-1}, \sigma_g^{m-1}, Y_i = g) q_m(\theta_i^* \mid \theta_i^{m-1})}. \tag{8.5}$$

- Accept each $\theta_i^m = \theta_i^*$ with probability α_i^*; otherwise let $\theta_i^m = \theta_i^{m-1}$.

8.2.3 Steps to Sample Parameters for the Proficiency Distribution

Fix Population. The population parameters for each group $\lambda_g = (\mu_g, \sigma_g)$, $g = 1, 2, \cdots, G$ are estimated in BMIRT except one group for the reason of fixing the scale. Normally, the population for the middle grade is fixed to be normal or multi-normal, with mean 0 and variance-covariance matrix to be identity. Also the item discrimination parameters for that population has the following form

$$\sigma = \begin{pmatrix} * & 0 & 0 & \cdots \\ & * & 0 & \cdots \\ \vdots & \vdots & \ddots & \\ & * & \cdots & * \end{pmatrix}_{DxD} \tag{8.6}$$

Fix Item. The population parameters for each group $\lambda_g = (\mu_g, \sigma_g)$, $g = 1, 2, \cdots, G$ are estimated in BMIRT except one group of fixing the mean to be zero. Also fix the item discrimination parameters for that population has the following form

$$\sigma = \begin{pmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \cdots & 1 \end{pmatrix}_{DxD} \tag{8.7}$$

- Sample the variance-covariance matrix σ_g from $Inv - Wishart_{\nu_{N_g}}(\Lambda_{N_g}^{-1})$.
- Sample the mean $\mu_g \sim N(\overline{\theta}, \frac{\sigma_g}{N_g})$ for each $g = 1, 2, \cdots, G$, where N_g is the number in population gth group of examinees, and

$$\nu_{N_g} = N_g - 1, \tag{8.8}$$

$$\Lambda_{N_g} = \sum_{i=1}^{N_g} (\theta_i^m - \overline{\theta})^T (\theta_i^m - \overline{\theta}), \tag{8.9}$$

$$\overline{\theta} = \frac{1}{N_g} \sum_{i=1}^{N_g} (\theta_i^m \mid Y_i = g). \tag{8.10}$$

Note: μ and σ are not independent, but the weight of the prior of σ to the mean is almost zero.

8.2.4 Prior and Proposal Functions

Priors for the items. For multiple choice items (M-3PL model), we assume the following priors:

$$\beta_{1,j} \sim N(\mu_{\beta_{1,j}}, \sigma^2_{\beta_{1,j}}), \tag{8.11}$$

$$\log(\beta_{2,j,l}) \sim N(\log(\mu_{\beta_{2,j}}), \sigma^2_{\beta_{2,j}}), \tag{8.12}$$

for $l = 1, \cdots, D$

$$\beta_{3,j} \sim beta(a, b). \tag{8.13}$$

Also assume

$$P(\beta_j) = P(\beta_{1,j}) \prod_{l=1}^{D} P(\beta_{2,j,l}) P(\beta_{3,j}). \tag{8.14}$$

For constructed response items (M-2PPC model), the priors are taken to be lognormal for $\beta_{2,j,l}$, $l = 1, \cdots, D$ and normal for $\beta_{\delta_k,j}$, $k = 2, \ldots, K_j$.

Priors for the Proficiency. The priors for the proficiency is normal or multi-normal.

$$P(\theta_g \mid \mu_g, \sigma_g) \sim N(\mu_g, \sigma_g). \tag{8.15}$$

for $g = 1, 2, \cdots, G$. Noninformative priors for μ_g and σ_g are used.

Proposals for the items. For multiple choice items (M-3PL model), we assume the following proposal functions:

$$q_m(\beta_{1,j} \mid \beta_{1,j}^{m-1}) = \frac{1}{\sqrt{2\pi} C_{\beta_{1,j}}} e^{-\frac{(\beta_{1,j} - \beta_{1,j}^{m-1})^2}{2C_{\beta_{1,j}}^2}}, \tag{8.16}$$

$$q_m(\beta_{2,j,l} \mid \beta_{2,j}^{m-1}) = \frac{1}{\sqrt{2\pi} C_{\beta_{2,j}}} e^{-\frac{(\log \beta_{2,j} - \log \beta_{2,j}^{m-1})^2}{2C_{\beta_{2,j}}^2}} \frac{1}{\beta_{2,j}}, \tag{8.17}$$

for $l = 1, \cdots, D$

$$q_m(\beta_{3,j} \mid \beta_{3,j}^{m-1}) = \frac{1}{2\delta} 1_{(\beta_{3,j}^{m-1} - \delta, \beta_{3,j}^{m-1} + \delta)}(\beta_{3,j}), \tag{8.18}$$

where

$$\left\{ \begin{array}{cc} 1_{(\beta_{3,j}^{m-1} - \delta, \beta_{3,j}^{m-1} + \delta)}(\beta_{3,j}) = 1 & \text{if } \beta_{3,j} \in (\beta_{3,j}^{m-1} - \delta, \beta_{3,j}^{m-1} + \delta) \\ 0 & \text{otherwise} \end{array} \right\}$$

To compute the acceptance rate, we see that:

$$q_m(\beta_j^* \mid \beta_j^{m-1}) = q_m(\beta_{1,j}^* \mid \beta_{1,j}^{m-1}) \prod_{l=1}^{D} q_m(\beta_{2,j,l}^* \mid \beta_{2,j,l}^{m-1}) q_m(\beta_{3,j}^* \mid \beta_{3,j}^{m-1}), \tag{8.19}$$

and

$$q_m(\beta_j^{m-1} \mid \beta_j^*) = q_m(\beta_{1,j}^{m-1} \mid \beta_{1,j}^*) \prod_{l=1}^{D} q_m(\beta_{2,j,l}^{m-1} \mid \beta_{2,j,l}^*) q_m(\beta_{3,j}^{m-1} \mid \beta_{3,j}^*). \tag{8.20}$$

So

$$\frac{q_m(\beta_j^{m-1} \mid \beta_j^*)}{q_m(\beta_j^* \mid \beta_j^{m-1})} = \frac{\prod_{l=1}^{D} q_m(\beta_{2,j,l}^{m-1} \mid \beta_{2,j,l}^*)}{\prod_{l=1}^{D} q_m(\beta_{2,j,l}^* \mid \beta_{2,j,l}^{m-1})} \tag{8.21}$$

$$= \frac{\prod_{l=1}^{D} \beta_{2,j,l}^*}{\prod_{l=1}^{D} \beta_{2,j,l}^{m-1}}. \tag{8.22}$$

For constructed response items (M-2PPC model), the proposal functions are taken to be lognormal for $\beta_{2,j,l}$, $l = 1, \cdots, D$, and normal for $\beta_{\delta_k,j}$, $k = 2, \ldots, K_j$.

Proposals for the Proficiency. Multivariate normal functions are used for the proposal of θ, that is

$$q_m(\theta \mid \theta^{m-1}) \sim N(\theta^{m-1}, \sigma_\theta). \tag{8.23}$$

8.3 MIRT Ability Estimation Methods and Standard Error of Measurement (SEM)

Ability estimates in the MIRT framework can be produced by Bayesian or non-Bayesian. Similar to the unidimensional IRT (UIRT), there are three methods that can be used to estimate abilities in the MIRT framework. They are: (a) MLE: Maximum likelihood estimation methods; (b) MAP: Maximize a posterior; (c) EAP, Expected a posterior. With the development of MCMC technique, the abilities can be derived by MCMC sampling for the posterior distribution and the mean of the ability samplings after the burn-in are their estimates; this is similar to EAP. For MAP and EAP, strong priors, standard normal, and noninformative priors can be applied. The following statistics will be used in computing the estimates and their standard error of measurement.

8.3.1 Statistics

First-Derivative.

$$\frac{\partial \log L(\vec{X} \mid \vec{\theta})}{\partial \vec{\theta}} = \sum_{j=1}^{J} \frac{\partial \log P_j}{\partial \vec{\theta}}, \tag{8.24}$$

where

$$\frac{\partial \log P_j}{\partial \vec{\theta}} = \sum_{k=1}^{K_j} 1_{(X_j=k-1)} \frac{\partial \log P_{jk}}{\partial \vec{\theta}} = \sum_{k=1}^{K_j} 1_{(X_j=k-1)} \frac{\partial P_{jk}}{\partial \vec{\theta}} \frac{1}{P_{jk}} = \sum_{k=1}^{K_j} 1_{(X_j=k-1)}((k-1)-E_j)\vec{\beta}_{2j}, \tag{8.25}$$

and

$$E_j = \sum_{k=1}^{K_j}(k-1)P_{jk} = \frac{\sum_{k=1}^{K_j}(k-1)e^{(k-1)\vec{\beta}_{2j}\odot\vec{\theta}^T - \sum_{t=1}^{k}\beta_{\delta_t j}}}{\sum_{m=1}^{K_j} e^{(m-1)\vec{\beta}_{2j}\odot\vec{\theta}^T - \sum_{t=1}^{m}\beta_{\delta_t j}}}, \tag{8.26}$$

for an M-2PPC item. For an M-3PL item,

$$\frac{\partial \log P_j}{\partial \vec{\theta}} = (\frac{1_{(X_j=1)}}{P_{j1}} - \frac{1_{(X_j=0)}}{1-P_{j1}})\frac{\partial P_{j1}}{\partial \vec{\theta}} = \frac{(X_j - P_{j1})(P_{j1} - \beta_{3j})}{P_{j1}(1-\beta_{3j})}\vec{\beta}_{2j}. \tag{8.27}$$

Second-Derivative.

$$J(\vec{\theta}) = \frac{\partial^2 \log L(\vec{X} \mid \vec{\theta})}{\partial \vec{\theta}^2} = \sum_{j=1}^{J} \frac{\partial^2 \log P_j}{\partial \vec{\theta}^2}, \tag{8.28}$$

and

$$\frac{\partial^2 \log P_j}{\partial \vec{\theta}^2} = -\sum_{k=1}^{K_j} 1_{(X_j=k-1)} \frac{\partial E_j}{\partial \vec{\theta}} \otimes \vec{\beta}_{2j} = -\sigma_j^2 \vec{\beta}_{2j} \otimes \vec{\beta}_{2j}, \tag{8.29}$$

where

$$\sigma_j^2 = \sum_{k=1}^{K_j}(k-1)^2 P_{jk} - E_j^2, \tag{8.30}$$

for an M-2PPC item. Here $\otimes$ is a vector product; $\vec{\beta}_{2j} \otimes \vec{\beta}_{2j}$ is a $D \times D$ matrix, and its mth row and nth column element is the product of the mth and nth element of $\vec{\beta}_{2j}$. For an M-3PL item j,

$$\frac{\partial^2 \log P_j}{\partial \vec{\theta}^2} = \frac{\beta_{3j}X_j - P_{j1}^2}{P_{j1}^2} \frac{1}{1-\beta_{3j}} \frac{\partial P_{j1}}{\partial \vec{\theta}} \otimes \vec{\beta}_{2j} = \frac{(1-P_{j1})(P_{j1}-\beta_{3j})(\beta_{3j}X_j - P_{j1}^2)}{P_{j1}^2(1-\beta_{3j})^2}\vec{\beta}_{2j} \otimes \vec{\beta}_{2j}. \tag{8.31}$$

Item and Test Information Function. For an item j, following M-3PL, the information function at $\vec{\theta}$ is

$$I_j(\vec{\theta}) = -E\frac{\partial^2 \log P_j}{\partial \vec{\theta}^2} = \frac{(P_{j1}-\beta_{3j})^2(1-P_{j1})}{P_{j1}(1-\beta_{3j})^2}\vec{\beta}_{2j} \otimes \vec{\beta}_{2j}. \tag{8.32}$$

For an item j, following M-2PPC, the information function at $\vec{\theta}$ is

$$I_j(\vec{\theta}) = \sigma^2 \vec{\beta}_{2j} \otimes \vec{\beta}_{2j}, \tag{8.33}$$

where $\sigma^2 = \sum_{k=1}^{K_j}(k-1)^2 P_{jk} - (\sum_{k=1}^{K_j}(k-1)P_{jk})^2$. The test information for J items at $\vec{\theta}$ is $\mathbf{I}_J(\vec{\theta}) = \sum_{j=1}^{J} I_j(\vec{\theta})$. The directional information in the direction $\vec{\alpha}$ is $\cos(\vec{\alpha})\mathbf{I}_J\cos(\vec{\alpha})^T$, where $\cos(\vec{\alpha}) = (\cos\alpha_1, \cdots, \cos\alpha_D)$ and α_l is the angle between $\vec{\theta}$ and θ_l.

8.3.2 Bayesian Statistics

Suppose the prior of population is $f(\vec{\theta})$, then the posterior density function of $\vec{\theta}$ is

$$f(\vec{\theta} \mid \vec{X}) \propto L(\vec{X} \mid \vec{\theta}) f(\vec{\theta}), \tag{8.34}$$

and if the prior is normal $N(\vec{\mu}, \Sigma)$, then

$$f(\vec{\theta}) = (2\pi)^{-D/2}(\mid \Sigma \mid)^{-1/2} exp(-\frac{1}{2}(\vec{\theta} - \vec{\mu})^T \Sigma^{-1}(\vec{\theta} - \vec{\mu})), \tag{8.35}$$

where $\vec{\mu}$ and Σ represent the population mean and variance-covariance matrix, respectively.

First-Derivative.

$$\frac{\partial \log f(\vec{\theta} \mid \vec{X})}{\partial \vec{\theta}} = \frac{\partial \log L(\vec{X} \mid \vec{\theta})}{\partial \vec{\theta}} - \frac{\partial(\vec{\theta} - \vec{\mu})}{\partial \vec{\theta}} \Sigma^{-1}(\vec{\theta} - \vec{\mu}), \tag{8.36}$$

where

$$\frac{\partial(\vec{\theta} - \vec{\mu})}{\partial \theta_k} = (0, \cdots, 1, 0, \cdots, 0)_{1\times D},$$

and 1 is in the kth position.

Second-Derivative.

$$\frac{\partial^2 \log f(\vec{\theta} \mid \vec{X})}{\partial \vec{\theta}^2} = \frac{\partial^2 \log L(\vec{X} \mid \vec{\theta})}{\partial \vec{\theta}^2} - \Sigma^{-1} = J(\vec{\theta}) - \Sigma^{-1}. \tag{8.37}$$

Item and test information function. Posterior test information at $\vec{\theta}$ for selected $j-1$ items is

$$\mathbf{I}_{j-1}(\vec{\theta}) = -EJ(\vec{\theta}) + \Sigma^{-1} = -\sum_{j=1}^{J} E\frac{\partial^2 \log P_j}{\partial \vec{\theta}^2} + \Sigma^{-1}. \tag{8.38}$$

Here bold variable $\mathbf{I}_{j-1}$ indicates the sum of $I_1, \cdots, I_{j-1}$.

The ability estimation methods are described below.

- MLE: MIRT ability is estimated by finding the mode $\hat{\vec{\theta}}$ that maximize the likelihood function $L(\vec{X} \mid \vec{\theta})$, i.e.,

$$\frac{\partial \log L(\vec{X} \mid \vec{\theta})}{\partial \vec{\theta}} \mid \hat{\vec{\theta}} = 0. \tag{8.39}$$

  Using Newton-Raphson method, suppose $\vec{\theta}^m$ is the m-th approximation that maximize $\log L(\vec{X} \mid \vec{\theta})$, then

$$\vec{\theta}^{m+1} = \vec{\theta}^m - \vec{\delta}^m, \tag{8.40}$$

  where

$$\vec{\delta}^m = [\mathbf{J}(\vec{\theta}^m)]^{-1} \times \frac{\partial \log L(\vec{X} \mid \vec{\theta})}{\partial \vec{\theta}}, \tag{8.41}$$

  and $\mathbf{J}(\vec{\theta})$ is the matrix of the second partial derivative.

- MAP: MIRT ability is estimated by finding the mode that maximize the posterior likelihood function $f(\vec{\theta} \mid \vec{X})$. This is similar to MLE method, but the function used is the product of the likelihood and the prior–instead of the likelihood.

  MAP estimates are derived using Equation (23) and (24) for the Bayesian versions.

- EAP: Suppose there are Q quarture points in the θ range $(-3, 3)$ that forms the D-dimensional vector score point $\vec{\theta}_k$, where $k = 1, \cdots, Q^D$. The marginal likelihood function is

$$\int L(\vec{X} \mid \vec{\theta}) f(\vec{\theta}) d\theta_1 \cdots \theta_D = \sum_{k=1}^{Q^D} L(\vec{X} \mid \vec{\theta}_k) f(\vec{\theta}_k). \tag{8.42}$$

  The expectation is a vector of dimension D, and it is

$$E(\vec{X}) = \int L(\vec{X} \mid \vec{\theta}) f(\vec{\theta}) \vec{\theta} d\theta_1 \cdots \theta_D = \sum_{k=1}^{Q^D} L(\vec{X} \mid \vec{\theta}_k) f(\vec{\theta}_k) \vec{\theta}_k, \tag{8.43}$$

  where the lth componet of the left hand side is corresponding to the lth component of $\vec{\theta}_k$. The EAP estimates for $\vec{\theta}$ is

$$\hat{\vec{\theta}} = \frac{E(\vec{X})}{\int L(\vec{X} \mid \vec{\theta}) f(\vec{\theta}) d\theta_1 \cdots \theta_D}. \tag{8.44}$$

For MLE, it is to find $(\hat{\theta_1}, \cdots, \hat{\theta_D})$ such that $\prod_{j=1}^{J} P_j(X_j \mid \vec{\theta}, \vec{\beta}_j) = \prod_{l=1}^{D} \prod_{j \in O_l} P_j(X_j \mid \vec{\theta}, \vec{\beta}_j)$ has the maximum value, which is equivalent to find $\hat{\theta}_l$ that maximizing the likelihood for domain l for all $l = 1, \cdots, D$. Here O_l contains all the items in domain l. For Bayesian methods with standard normal as the prior, MAP or EAP for deriving $(\hat{\theta_1}, \cdots, \hat{\theta_D})$ is the same as the MAP or EAP for finding $\hat{\theta}_l$ with $N(0,1)$ as the prior when the items in the test are simple structured, where $l = 1, \cdots, D$, as the joint prior for $\vec{\theta}$ is the product of the priors for each of the domains. When the noninformative prior is used, for example, with variance $var = 10$, and covariance $cov = 0$, the Bayesian methods should yield similar results as those from MLE. Using standard normal or noninformative prior would ignore the correlated information between domains, while using strong priors would allow the information to be borrowed from each other and increase the precision, especially when the test is short. Therefore, compared to the ability estimates from UIRT, the ability estimates from MIRT would have better precision using MAP or EAP when the prior is strong and is neither standard normal nor noninformative (Yao & Boughton, 2007). Such a prior can be derived using the student raw data.

8.3.3 Composite Score and SEM

For a test with J items of known item parameters, for a given score point $\vec{\theta}$, the test information is $\mathbf{I}_J(\vec{\theta})$. the composite score $\theta_{\vec{\alpha}} = \sum_{l=1}^{D} \theta_l w_l$ has a standard error of measurement $SEM(\theta_{\vec{\alpha}}) = V(\theta_{\vec{\alpha}})^{1/2}$, where $V(\theta_{\vec{\alpha}}) = \vec{w} V(\vec{\theta}) \vec{w}^T$, $\vec{w} = (w_1, \cdots, w_D) = (cos^2\alpha_1, \cdots, cos^2\alpha_D)$. $V(\vec{\theta})$ can be approximated by $I(\vec{\theta})^{-1}$. The SEM for each domain can be derived by using the angle for that dimension to be 0, and all other angles 90^0.

The composite score for the D-dimensinal domain scores can be obtained by simply averaging the domain scores with a set of predetermined weights, by the weighted sum of the domain scores and the optimized weight (Yao, 2011, 2012), or by higher-order MIRT (de la Torre & Hong, 2010; de la Torre & Song, 2009; Yao, 2010b) model using MCMC that simultaneously estimate the domain abilities and the composite abilities. For the optimized weight, it yields the composite score with the smallest SEM, and therefore has a better prediction or estimation for an examinee's overall ability. The optimized weight can be derived by formula (Yao, 2012) or by computer program (BMIRT, 2010).

8.4 Computation of Model Fit Statistics

The following fit statistics are used to evaluate the model fit.

χ^2. Let $df_m = (m+2)J_1 + \sum_{j=1}^{J_2}(K_j + m - 1) + m \times N$, where m is the number of dimension, N is the number of examinees, J_1 is the number of multiple choice items, and J_2 is the number of polytomously-scored items. The likelihood function for m- dimension denoted by $L_m = L(\mathbf{X} \mid \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}})$, then define

$$\chi^2_{m+1} = [2 \times logL_{m+1} - 2 \times logL_m]/[df_{m+1} - df_m]. \tag{8.45}$$

AIC. $AIC_m = -2logL_m + 2df_m$.

DIC. The Bayesian deviance information criterion (DIC) introduce by Spiegelhalter, Best, Carlin, and van der Linde (2002) is defined as $DIC = \overline{D} + p_D$, where $\overline{D} = -2E(\log L(\mathbf{X} \mid \boldsymbol{\theta}, \boldsymbol{\beta})$ and $p_D = \overline{D} - 2\log(L(\mathbf{X} \mid \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}))$ is th effective number of parameters. For each MCMC iteration $l = 1, 2, \cdots L$, with the sampling of the parameters denoted by $(\boldsymbol{\theta}^l, \boldsymbol{\beta}^l)$, $\overline{D} = \frac{-2\sum_{l=1}^{L} \log L(X|\boldsymbol{\theta}^l, \boldsymbol{\beta}^l)}{L}$.

$$BIC_k = G_k^2 + 2\log(N)df_k. \tag{8.46}$$

8.5 Score Distribution

Let S_J denotes the summed score for a total of J items. The conditional distribution of $S_J = s$ for an examinee of ability $\vec{\theta} = (\theta_1, \cdots, \theta_D)$ denoted by $P_s(\vec{\theta}) = P(S_J = s \mid \vec{\theta})$ is calculated using a recursion formula suggested by Lord and Wingersky (1984), and $s = 0, 1, \cdots, T$, where T is the total score of the test, which is obtained by taking the summation of the maximum score points of all the items in the test (for a M-3PL item, the maximum score is

1; for a M-2PPC item, the maximum score is the level or category of the item minus 1). The following described the recursion formula for the MIRT model and for items of mixed item types:

Let X_j be a random variable representing the score on item j. X_j can take values of $0, 1, \cdots, K_j - 1$, where $K_j = 2$ for an item j following M-3PL. It is obvious that

$$S_J = \sum_{j=1}^{J} X_j. \tag{8.47}$$

Therefore,

$$\begin{aligned} P_s(\vec{\theta}) = P(S_J = s \mid \vec{\theta}) = P(S_{J-1} + X_J = s \mid \vec{\theta}) &= \sum_{x=0}^{K_J-1} P(S_{J-1} = s - x, X_J = x \mid \vec{\theta}) \\ &= \sum_{x=0}^{K_J-1} P(S_{J-1} = s - x \mid \vec{\theta}) P(X_J = x \mid \vec{\theta}), \end{aligned} \tag{8.48}$$

since S_{J-1} and X_J are independent for a given ability. Please note that $P(X_J = x \mid \vec{\theta})$ is defined by Equation 1 and 2, respectively, following M-3PL and M-2PPC.

8.6 Classification Consistency and Accuracy

Suppose examinees are classified into M categories, with $s_1, s_2, \cdots, s_{M-1}$ as the summed cut scores. Let $s_0 = 0$, the minimum score, and $s_M = T$, the maximum score. The probability that an examinee with ability $\vec{\theta}$ is classified to be category m is

$$p_{\vec{\theta}}(m) = \sum_{s=s_{m-1}}^{s_m} P_s(\vec{\theta}), \tag{8.49}$$

where $m = 1, 2, \cdots, M$.

The *conditional classification consistency index*, $\phi_{\vec{\theta}}$, is defined as the probability that an examinee with ability $\vec{\theta}$ is classified into the same category on independent administrations of two parallel forms of a test, and it can be computed as

$$\phi_{\vec{\theta}} = \sum_{m=1}^{M} [p_{\vec{\theta}}(m)]^2. \tag{8.50}$$

The *marginal classification consistency index* ϕ is given by

$$\phi = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \phi_{\vec{\theta}} f(\vec{\theta}) d\theta_1 \cdots d\theta_D, \tag{8.51}$$

where $f(\vec{\theta})$ is the density of multivariate normal distribution.

If the cut scores are expressed by θ metric for the composite/overall ability or the unidimensional IRT ability, summed cut scores need to be obtained in order to compute the *conditional classification consistency index*. Let the overall score be classified into M category, with cut scores $\theta_1, \cdots, \theta_{M-1}$. Now we transform the $M-1$ θ metric score $\theta_1, \cdots, \theta_{M-1}$ into $M-1$ summed score $s_1, \cdots, s_{M-1}$. Suppose there are J_1 M-3PL items and J_2 M-2PPC items, then

$$s_m = \sum_{j \in 3PL}^{J_1} P(X_j = 1 \mid \theta = \theta_m) + \sum_{j \in 2PPC}^{J_2} \sum_{k=0}^{K_j - 1} kP(X_j = k \mid \theta = \theta_m), \tag{8.52}$$

for $m = 1, \cdots, M-1$. Let $s_0 = 0$ and $s_M = T$. The item parameters in Equation 11 is obtained from the unidimensional IRT calibration. Large errors might be observed in transforming θ_m to s_m using the unidimensional IRT model if the data is actually multidimensional; further study about these effects needs to be conducted. The current study will only focus on using the summed cut scores.

If the cut scores are expressed by $\vec{\theta}$ metric for the multidimensional ability, then the following formula will transfer $\vec{\theta}_m$ to s_m.

$$s_m = \sum_{j \in 3PL}^{J_1} P(X_j = 1 \mid \vec{\theta}_m) + \sum_{j \in 2PPC}^{J_2} \sum_{k=0}^{K_j - 1} kP(X_j = k \mid \vec{\theta}_m). \tag{8.53}$$

The *conditional false positive error rate* is the probability that an examinee is classified into a category that is higher than the examinees's true category. The *conditional false negative error rate* is the probability that an examinee is classified into a category that is lower than the examinees's true category. For an examinee with ability $\vec{\theta}$, obtain the expected summed score by

$$\tau_{\vec{\theta}} = \sum_{j \in 3PL}^{J_1} P(X_j = 1 \mid \vec{\theta}) + \sum_{j \in 2PPC}^{J_2} \sum_{k=0}^{K_j - 1} kP(X_j = k \mid \vec{\theta}). \tag{8.54}$$

Two situations are possible:

- Suppose the true cut scores by summed score are known. The summed score $\tau_{\vec{\theta}}$ for the examinee with ability $\vec{\theta}$ is computed by Equation 13, then by comparing $\tau_{\vec{\theta}}$ with the true cut scores, we know the classification $t_{\vec{\theta}}$ for this examinee. The conditional probability of accurate classification is $p_{\vec{\theta}}(t_{\vec{\theta}})$. The marginal classification accuracy is

$$\gamma = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p_{\vec{\theta}}(t_{\vec{\theta}}) f(\vec{\theta}) d\theta_1 \cdots d\theta_D. \tag{8.55}$$

  The *conditional false positive error rate* is obtained by

$$\gamma_{\vec{\theta}}^{+} = \sum_{m=t_{\vec{\theta}}+1}^{M} p_{\vec{\theta}}(m). \tag{8.56}$$

  The *conditional false negative error rate* is obtained by

$$\gamma_{\vec{\theta}}^{+} = \sum_{m=0}^{t_{\vec{\theta}}-1} p_{\vec{\theta}}(m). \tag{8.57}$$

  The *marginal false positive error rate (FP)* and the *marginal false negative error rate(FN)* are given by

$$\gamma^{+} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \gamma_{\vec{\theta}}^{+} f(\vec{\theta}) d\vec{\theta} \tag{8.58}$$

$$\gamma^{-} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \gamma_{\vec{\theta}}^{-} f(\vec{\theta}) d\vec{\theta} \tag{8.59}$$

- Suppose the true cut scores by $\vec{\theta}$ metric are known and are denoted by $\vec{\theta}_1, \cdots, \vec{\theta}_{M-1}$. Using Equation 13, we can derive the corresponding summed cut score $s_1, \cdots, s_{M-1}$. Let $s_0 = 0$ and $s_M = T$. Then the above procedure repeat. Please note that the map between $\vec{\theta}$ and the expected score by Equation 13 is not one-to-one; for the compensatory MIRT model, different $\vec{\theta}$ may give the same expected score. Therefore, caution has to be taken when interpreting the cut scores in the multidimensional $\vec{\theta}$ metric.

For all the integrals introduced for both accuracy and consistency, two approaches are possible (Lee, 2010). If the integral is taken over the population distribution, then the results are called D method; if the integral is taken over the average of all the examinees in the data set, then the results are called P method.

8.7 Domain Score

Domain scores from BMIRT were obtained by the following: For student i, the domain score D_l for objective l, expected percentage of points for objective l for the test was obtained by

$$D_l = \frac{\sum_{j \in O_l, j \in M-2PPC} \sum_{k=1}^{K_j} (k-1) P_{ijk}(x_{ij} = k-1 \mid \vec{\theta}_i, \vec{\beta}_j) + \sum_{j \in O_l, j \in M-3PL} P_{ij1}(x_{ij} = 1 \mid \vec{\theta}_i, \vec{\beta}_j)}{\sum_{j \in O_l, j \in M-2PPC} (K_j - 1) + \sum_{j \in O_l, j \in M-3PL} 1} \tag{8.60}$$

where O_l contains the items that contribute to the objective l. The polytomous items are specified as M-2PPC and multiple-choice as M-3PL in the above expression.

Chapter 9

References

Yao, L. (2003). *BMIRT: Bayesian multivariate item response theory.* [Computer software]. Monterey, CA: Defense Manpower Data Ceneter.

Yao, L. (2004). *LinkMIRT: Linking of multivariate item response model.* [Computer software]. Monterey, CA: Defense Manpower Data Ceneter.

Yao, L. (2005). *An investigation of scaling options in estimating parameters for multidimensional item response theory model.* Unpublished manuscript.

Yao, L. (2010). Reoporting valid and reliability overall score and domain scores. *Journal of Educational Measurement. 47*, 339-360.

Yao, L. (2011). Multidimensional linking for domain scores and overall scores for nonequivalent groups. *Applied Psychological Measurement. 34*, 48-66.

Yao, L. (2012). Multidimensional CAT item selection methods for domain scores and composite scores: Theory and Applications. *Psychometrika, 2012, DOI: 10.1007/s11336-012-9265-5, 77(3)*, 495-523.

Yao, L. (2013). Comparing the performance of five multidimensional CAT selection procedures with different stopping rules. *Applied Psychological Measurement, 37*, 3-12.

Yao, L. (2014a). Multidimensional CAT item selection methods for domain scores and composite scores with item exposure control and content constraints. *Journal of Educational Measurement.*

Yao, L. (2014b). Multidimensional Item Response Theory for Score Reporting. In Chang & Cheng (Ed), Advances in Modern, International Testing: Transition from Summative to Formative Assessment. Charlotte, NC: Information Age Publishing.

Yao, L (2014c, April). Optimized Item Pool Generation and the Performance of Multidimensional CAT. Invited Presentation at the 2014 meeting of the National Council on Measurement in Education Philadelphia, Pennsylvania.

Yao, L., & Boughton, K. A. (2007). A Multidimensional item response modeling approach for improving subscale proficiency estimation and classification. *Applied Psychological Measurement. 31*, 83-105.

Yao, L., & Bough, K. A. (2009). Multidimensional linking for tests containing polytomous items. *Journal of Educational Measurement. 46*, 177-197.

Yao. L., Bough, K.,& Lewis, D. (2006, April). *Reporting Subscale Scores for Tests Composed of Complex Structure.* Paper presented at the annual meetings of the National Council on Measurement in Education, San Francisco, CA.

Yao, L, Lewis, D., & Zhang, L. (2008, April). *An introduction to the application of BMIRT: Bayesian multivariate item response theory software.* Training secession presented at the annual meetings of the National Council on Measurement in Education, NY.

Yao, L., Patz, R., & Lewis, D., (2003, April). *Multidimensional IRT Models Applied to Vertical Scaling.* Paper presented at the Annual Meeting of National Council on Measurement in Education, Chicago,IL.

Yao, L. Pommerich, M, & Segall, D. (2014). Using Multidimensional CAT Incoorporating Item Response Time to Administer a Short, Yet Precise Screening Test. *Applied Psychological Measurement,*

Yao,L. Reckase, M, Yuan, H, & Cheng, Y (2012, April). *Multidimensional Item Response Theory: Theory and Applications and BMIRT, LinkMIRT, and SimuMIRT Software.* Training presented at the 2012 meeting of the National Council on Measurement in Education, Vancouver, Canada.

Yao, L., & Mao, X.,(2004, April). *Unidimensional and Multidimensional Estimation of Vertical Scaled Tests with Complex Structure.* Paper presented at the annual meetings of the National Council on Measurement in Education, San Diego, CA.

Yao, L., & Richard C, (2008, March). *Application of Testlet-Effect Model to Performance Assessments with Multiple-Criteria Scoring Rubrics.* Paper presented at the annual meetings of the National Council on Measurement in Education, NY.

Yao, L., & Schwarz, R.D. (2006). A multidimensional partial credit model with associated item and test statistics: an application to mixed-format tests. *Applied Psychological Measurement. 30*, 469-492.

Yao,L. & Schwarz, R (2014, April). Comparsion of Methods in Detecting the Number of Dimensions and Item Clusters for Mixed Format and Mixed Structured Data Using MCMC Estimates. Paper Presented at the 2014 meeting of the National Council on Measurement in Education Philadelphia, Pennsylvania.

Yao, L., & Li, F., (2010, May). *A DIF Detection Procedure in Multidimensional Framework and its Applications* Paper presented at the annual meetings of the National Council on Measurement in Education, Denver, CO.

Patz , R. J. & Yao, L. (2006). Vertical scaling: Statistical models for measuring growth and achievement. In Sinharay, S., and Rao, C. R. (Eds.), Handbook of Statistics. Volum 26, 955-975.

Patz , R. J. & Yao, L. (2007). Methods and Models for vertical Scaling:Linking and Aliging Scores and Scales. In Dorans, N.J. & Holland, P.W.(Eds.), Statistics for Social Science and Behavorial Sciences, 253-273.

Reckase, M. D. (2009). *Multidimensional Item Response Theory.* New York: Springer.

Grima, A. M., & Yao, L.(2011). *Classification Consistency And Accuracy: Unidimensional Versus Multidimensional IRT Procedures.* Paper presented at the annual meetings of the National Council on Measurement in Education, New Orleans, Louisiana.

Wang, J. & Yao, L (2011, April). *The Effects of Scoring Designs and Rater Severity on Students' Ability Estimation for Constructed Response Items.* Paper presented at the annual meetings of the National Council on Measurement in Education, New Orleans, Louisiana.

Wang, J.& Yao, L (2012, April). *The Effects of Rater Distributions and Rater Severity on Students' Ability Estimation for Constructed-Response Items.* Paper Presented at the 2012 meeting of the National Council on Measurement in Education, Vancouver, Canada.

Wang, J. & Yao, L (2013). The Effects of Rater Distributions and Rater Severity on Students' Ability Estimation for Constructed-Response items. Research Report, ETS RR-13-23. http://search.ets.org/researcher/.

Wei, H & Yao, L (2013). *A Comparison of IRT Linking and Trend Scoring in MixedFormat Test Equating.* Paper presented at the 2013 meeting of the National Council on Measurement in Education, San Francisco, CA.

Kim, Y.Y. (2011). *A MIRT Application to Inform Trend Decisions in NAEP.* Study Report, Washington, D.C.: NAEP Education Statistics Services Institute. December 2011.

Lin, P (2008). IRT vs. factore analysis approaches in analyzing multigroup multidimensional binary data: the effect of structural orthogonality, and the equivalence in test structure, item difficulty, and examinee groups. Doctor of Philosophy Dissertation at University of Maryland.

Liu, H. Y., Li, C., Zhang, P., & Luo, F. (2012). Testing Measurement Equivalence of Categorical Items' Threshold/Difficulty Parameters: A Comparison of CCFA and (M)IRT Approaches. *Acta Psychologica Sinica*, *44 (8)*, 1124-1136.