Item Factor Analysis:
A primer and new open-source implementation

Joshua Nathaniel Pritikin
University of Virginia

B.S. Psychology, University of Oregon, 2009

A Predissertation presented to the Graduate Faculty
of the University of Virginia in Candidacy for the Degree of
Master of Arts

Department of Psychology

University of Virginia
Dec, 2013

Abstract

Have you collected survey data? Here is how to analyze it and why. Interval scale data is obtained from nominal or ordinal data by partitioning data conditional on examinee skill. Item Factor Analysis (a.k.a. Item Response Theory) fits a model to these types of data. Underlying mathematics are presented interwoven with insights gleaned from the experience of writing an open-source software implementation from scratch. Simulation studies, with complete source code included, demonstrate the accuracy of parameter recovery.

Contents

$$\underbrace{\text{Given } 2+3,\ \text{circle the correct answer:}}_{\textit{stimulus}}\quad \underbrace{3\ \mid\ 5\ \mid\ 6}_{\textit{response}}$$

Figure 1. Anatomy of an item. Responses are nominal (this example) or ordinal.

$$\underbrace{\text{I love to eat broccoli.}}_{\textit{stimulus}}\quad \underbrace{\text{Agree}\ \mid\ \text{Not sure}\ \mid\ \text{Disagree}}_{\textit{response}}$$

Figure 2. Anatomy of an item. Responses are nominal or ordinal (this example).

Conceptual Development of Item Factor Analysis

Test designers hope to understand what items on a test tell us about persons, and conversely, what persons tell us about items. An item consists of a stimulus and response (see Figures 1 and 2). Of interest here are items with a response on a nominal or ordinal scale. Consider nominal items that are scored *correct* or *incorrect*. Some items are more difficult than others to answer correctly. We may have a guess about item difficulty before administering an item set to examinees. For example, calculus items are likely to be relatively more difficult than arithmetic items to get correct. However, item difficulty is also estimable from data using the Classical Test Theory (CTT) statistic *proportion correct* (Lord, Novick, & Birnbaum, 1968, p. 328). The meaning of proportion correct depends on the sample. If a sample includes a substantial proportion of inappropriate examinees, then obtained estimates of difficulty will be inaccurate for a larger population. For example, administering a calculus test to elementary school students is not a good way to obtain precise difficulty estimates for college students. In addition to proportion correct, CTT offers estimates of item discrimination (item-total correlation) and test reliability. However, CTT is no longer the sole way to analyze item response data. In order to trace the evolution of CTT to modern item analysis, consider the following simulation.

Suppose 500 random examinees are characterized by skill drawn from the uniform distribution $[-5, 5]$. For an item with a proportion correct of 0.4, there are 200 examinees who responded correctly and 300 who responded incorrectly. Presumably, examinees who responded correctly all had a higher skill than examinees who responded incorrectly. However, an assumption of CTT is that observed score = true score + error. To account for the error, we add some Gaussian noise to examinee skill.

$$skill_{observed} = skill_{true} + \mathcal{N}(0, precision^{-1})$$

Since true skill is known, we can order response outcome by it and partition the data into 20 evenly spaced bins. Responses outcomes from examinees with similar skill are aggregated into % correct (see Figure 3). Observe that nominal or ordinal data has been converted into interval scale data conditional on examinee skill. This is really remarkable!

Figure 4 shows how error variance (i.e., $precision^{-1}$) and percentage correct change the shape of plots of % correct conditional on skill. These plots all approximate the Normal cumulative distribution function (CDF). Thereby, we can take the CDF as an item response model. As a practical matter, the Normal CDF is used merely as the equation for a curve. Its statistical properties as a CDF are not important in this context. Historically, the

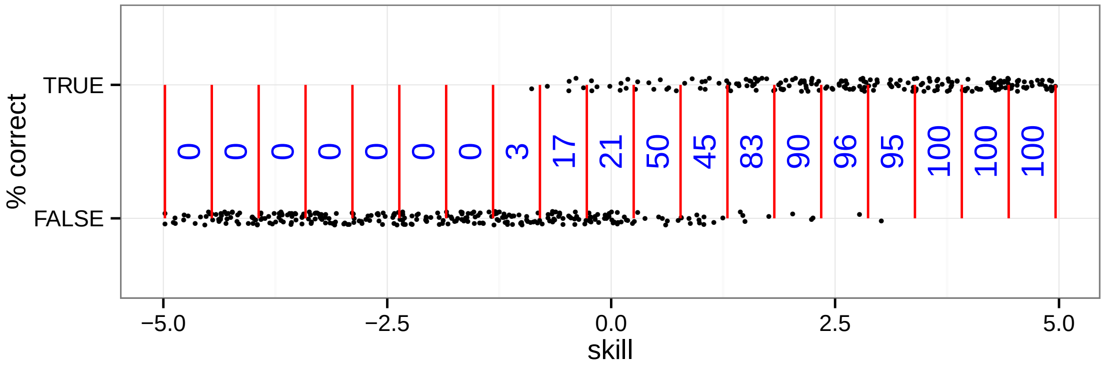

Figure 3. Percentage correct of responses by true skill bin (20 bins, in this example). This has the effect of converting nominal or ordinal data into interval scale data conditional on examinee skill.

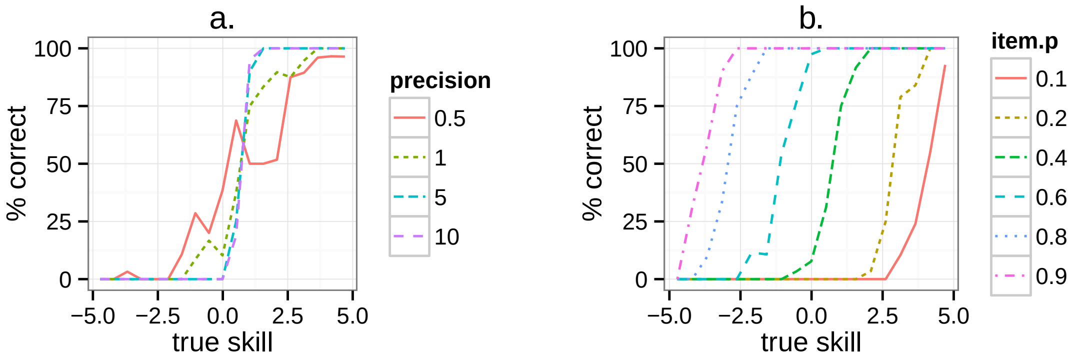

Figure 4. Percent correct per bin vs true skill for a variety of precisions (a) and proportions correct (b).

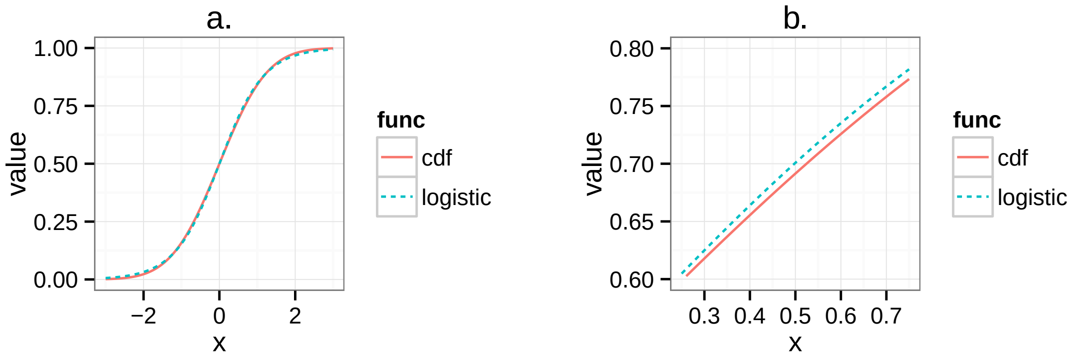

Figure 5. The Normal CDF and the logistic with a scaling constant of 1.702. In plot (a), the curves are almost too similar to distinguish. In plot (b), an area where the curves mismatch slightly is magnified to make the difference more visible.

Normal CDF was the model of choice for this kind of data. However, the Normal CDF is computationally inconvenient because it includes

$$\int_{-x}^{x} e^{-t^2}\ \mathrm{d}t,$$

which cannot be reduced further to elementary functions and is computationally expensive to approximate. A more convenient alternative is the logistic $\mathrm{P}(x) = \frac{1}{1+e^{-x}}$. With a scaling constant of 1.702, the two curves differ by less than .01 over the whole domain of interest (Camilli, 1994; also see Figure 5).

The observation that item response patterns can be represented by a model sparked a new analysis approach called Item Response Theory, Modern Test Theory, or Item Factor Analysis (IFA). Henceforth, the term IFA will be used to describe this class of methods. Additional parameters a and c are introduced in the logistic and become the focus of item characterization. For example, if an examinee with latent skill θ is given a 2 alternative forced choice to pick either response 0 or 1 then the model predictions are,

$$\mathrm{P}(\mathrm{pick} = 0|a, c, \theta) = 1 - \mathrm{P}(\mathrm{pick} = 1|a, c, \theta) \tag{1}$$

$$\mathrm{P}(\mathrm{pick} = 1|a, c, \theta) = \frac{1}{1 + \exp(-(a\theta + c))}. \tag{2}$$

Drawing from the tradition of logistic regression, a is the *slope* and c is the *intercept* (Lord et al., 1968, p. 353). In the tradition of IFA, Equation 2 is the 2PL item model. The a slope parameter is often called *discrimination* because higher values are characteristic of items that are more sensitive to examinee skill. The c intercept parameter does not have an intuitive interpretation. In the single factor case, $b = -\frac{c}{a}$ offers an intuitive index of item difficulty. The same idea can be extended to multidimensional items by using the norm $|a|$ in place of a (Reckase, 1985).

IFA and CTT, Similarities and Differences

In IFA, the difficulty of an item is conditional on examinee skill. For example, an easy item for a university student may be difficult for a 4th grader. In CTT, the proportion correct statistic is not explicitly conditional on examinee skill, but is interpreted with respect to a norm group. In practice, this difference is extremely important. Administering a measure to a norm group is typically a large and complex enterprise. It is always easier to administer a measure to individuals. That IFA accumulates information from individual measurements is a major practical advantage of IFA over CTT.

Embretson and Reise (2000, Chapter 2) detailed many more advantages of IFA over CTT as is summarized here. Examinee ability, as estimated by CTT, has a constant standard error (a.k.a. standard error of measurement) across the range of examinee ability. This assumption defies common sense in all but the most controlled situations. Is it plausible that a math test designed to assess the mathematical ability of 1st graders offers equally accurate ability estimates for 2nd graders? The effect of test length is also affected by this assumption. Is it plausible that a test designed for 1st graders and administered to 1st graders is less accurate than a much longer test designed for 5th graders and administered to 1st graders? IFA takes into account how well item difficulties match examinee skill.

CTT item characteristics are very sensitive to the sample used to obtain them. An unrepresentative sample will result in biased estimates. In comparison, IFA estimates are less sensitive to the shape of the latent distribution. Both CTT and IFA scores can be interpreted with respect to a norm group. However, only IFA scores have a direct meaning in relation to item difficulty. When an examinee's IFA score is equal to the item's difficulty then that examinee has a 50% chance of responding correctly to that item. IFA can handle mixed item formats (e.g., a mixture of true/false and multiple choice items) whereas CTT cannot. Frequently researchers are interested in the change of a trait estimate when the same test is administered and then administered again after an intervention. Calculating change scores using CTT is controversial (Cronbach & Furby, 1970). In contrast, estimating change scores is not problematic in an IFA context.

IFA Models

Item Models

Item models assign probabilities to outcome categories conditional on examinee skill and item parameters. Certain item models can only deal with dichotomous (i.e. correct/incorrect) items and some models are suited to polytomous multiple outcome items. Over the years, many item models have been proposed. Refer to Thissen and Steinberg (1986) for an item model taxonomy. Fortunately, the most popular item models can be represented with just three flexible item models: the 4PL (Loken & Rulison, 2010), the graded response model (Samejima, 1969), and the nominal model (Thissen, Cai, & Bock, 2010). Each model will be described in turn.

Let θ be participant skill, a be a slope, c be an intercept, g a pseudo-guessing parameter, u an upper bound, k an outcome category, and K the number of outcome categories. The 4PL model is

$$\mathrm{P}(\mathrm{pick} = 1|a, c, g, u, \theta) = g + (u - g)\frac{1}{1 + \exp(-(a\theta + c))}. \tag{3}$$

The g pseudo-guessing parameter models the situation where low ability examinees may accidentally guess the correct answer. Similarly, the u upper bound models the situation where high ability examinees occasionally answer incorrectly (see Figure 6). When the pseudo-guessing parameter g and the upper bound u are fixed to 0 and 1, respectively, then the 4PL model (Equation 3) is equivalent to the 2PL model (Equation 2 and Figure 7). One reason that the 2PL model is important is because it easily extends to more than two outcomes where the category response probability is the difference between two adjacent 2PL models (Figure 8). This is known as the graded response model (Samejima, 1969). The graded response model is often a good choice for ordinal data (e.g., Figure 2). To obtain the 2PL as a special case of the 4PL requires parameter restrictions, however, the 2 category graded response model is equivalent to the 2PL without parameter restrictions. This makes the graded model a convenient choice when the popular 2PL is desired.

Some items have more than 2 possible outcomes but without a natural ordering. For example, a fruit preference item might ask whether you prefer bananas, pomelo, or custard apples. The graded response model could work with these data, but some research questions can be better served by the nominal response model. The nominal response model is

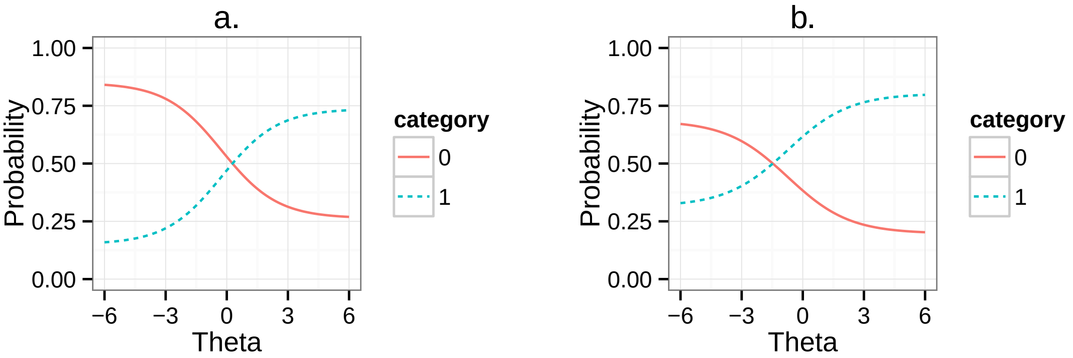

Figure 6. Two random single factor 4PL response plots. Parameters for plot (a) are $a = 0.73$, $c = 0.18$, $g = 0.15$, and $u = 0.74$; and for plot (b) are $a = 0.66$, $c = 0.49$, $g = 0.31$, and $u = 0.8$. For dichotomous items, plotting both categories 0 and 1 is redundent because $\mathrm{P}(\mathrm{pick} = 0) = 1 - \mathrm{P}(\mathrm{pick} = 1)$.

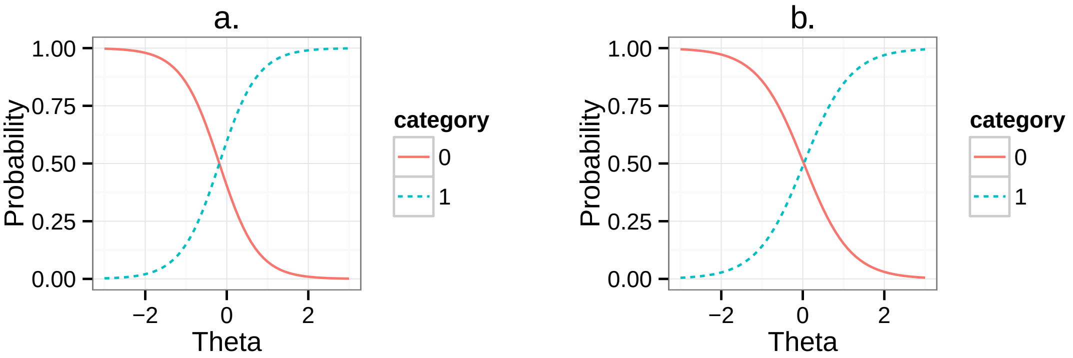

Figure 7. Two random single factor 2PL response plots. Parameters for plot (a) are $a = 2.13$ and $c = 0.39$, and for plot (b) are $a = 1.75$ and $c = -0.04$.

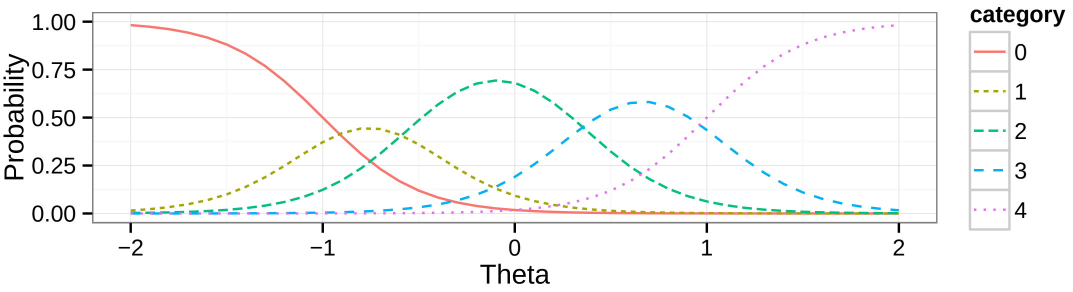

Figure 8. One single factor graded response probability plot, a=4 and $c_k = 4, 2.08, -1.33, -4$.

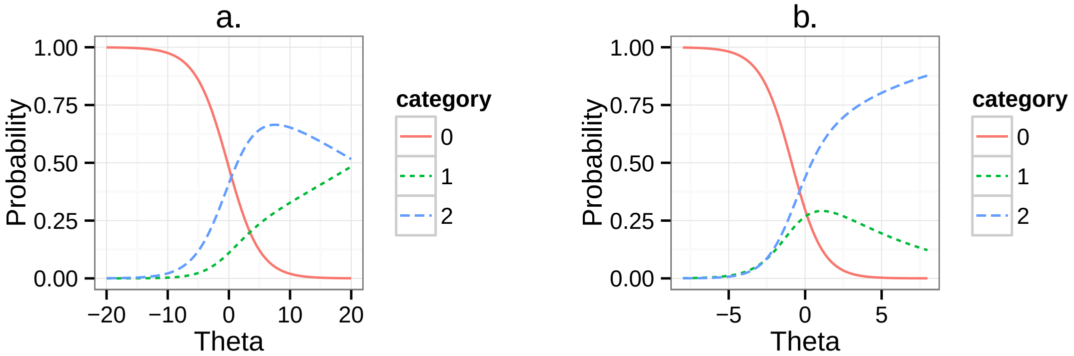

Figure 9. Two random single factor nominal model response plots. A Fourier basis (Equation 7) is used for both T_a and T_c. Typically there is a skill level for every category where that category is the most likely to be chosen. It so happened that, in both examples, category 1 is never the most likely category to be chosen. Observe that the probability for category 1 is always below some other category. In plot (a), category 1 is less likely than category 2 for $\theta > 4$ and less likely than both other categories for negative θ. The parameters for plot (a) are $a = 0.37$, $\alpha_1 = 0.49$, $\alpha_2 = 0.66$, $\gamma_1 = -0.08$, and $\gamma_2 = -1.39$; and for plot (b) are $a = 0.79$, $\alpha_1 = 0.67$, $\alpha_2 = 0.43$, $\gamma_1 = 0.19$, and $\gamma_2 = -0.3$. α and γ are estimated. The transformation matrices T_a and T_c are chosen by the analyst and not estimated.

$$a_k = T_a \alpha_k \tag{4}$$

$$c_k = T_c \gamma_k \tag{5}$$

$$\mathrm{P}(\text{pick} = k | a, a_k, c_k) = C \; \frac{1}{1 + \exp(-(a\theta a_k + c_k))} \tag{6}$$

where a_k and c_k are the result of multiplying two vectors of free parameters α and γ by fixed matrices T_a and T_c, respectively; a_0 and c_0 are fixed to 0 for identification; and C is a normalizing constant such that $\sum_k \mathrm{P}(\text{pick} = k) = 1$. Note that while the graded response model has a single slope parameter, the nominal response model can have a slope for every response outcome. The transformation matrices T_a and T_c make this model extraordinarily flexible because α and γ coefficients can be constrained to be fixed or equated to other parameters.

By default, a Fourier basis is used for the transformation matrices T_a and T_c,

$$\underset{(m-1)X(m-1)}{T_F} = \begin{bmatrix} 1 & f_{22} & \cdots & f_{2(m-1)} \\ 2 & f_{32} & \cdots & f_{3(m-1)} \\ \vdots & \vdots & & \vdots \\ m-1 & 0 & \cdots & 0 \end{bmatrix} \tag{7}$$

where the ith basis vector indexed by category k is

$$f_{ki} = \sin\left[\frac{\pi(i-1)(k-1)}{m-1}\right],$$

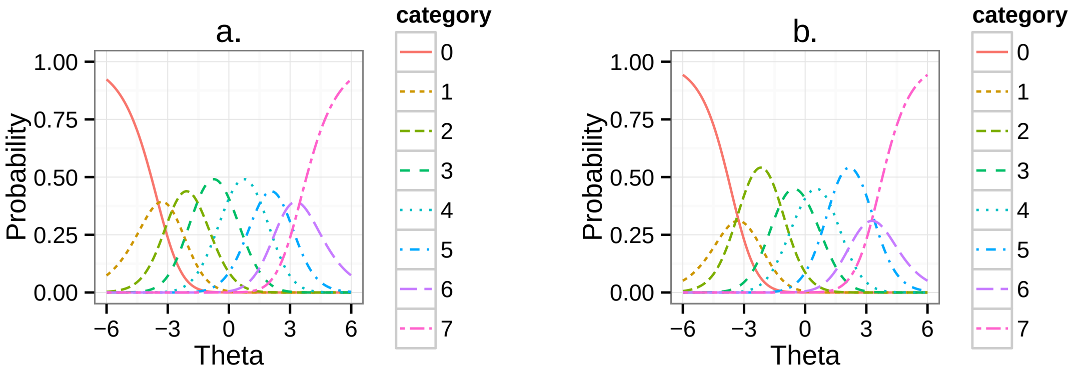

Figure 10. Two nominal response models parameterized using a Fourier basis. Plot (a) uses 1 for the linear coefficient of T_a and 8 for the 2nd basis vector of T_c. The remaining coefficients are set to 0. The 2nd basis vector of T_c spreads out the categories away from $\theta = 0$ to make a beautiful flower petal design. Plot (b) is the same as plot (a) except that the coefficient for the last basis vector of T_a is set to 0.15. This causes oscillation among the height of the category peaks. Using the Fourier basis, researchers can search for a balance between a versatile model and over-fitting with too many estimable parameters.

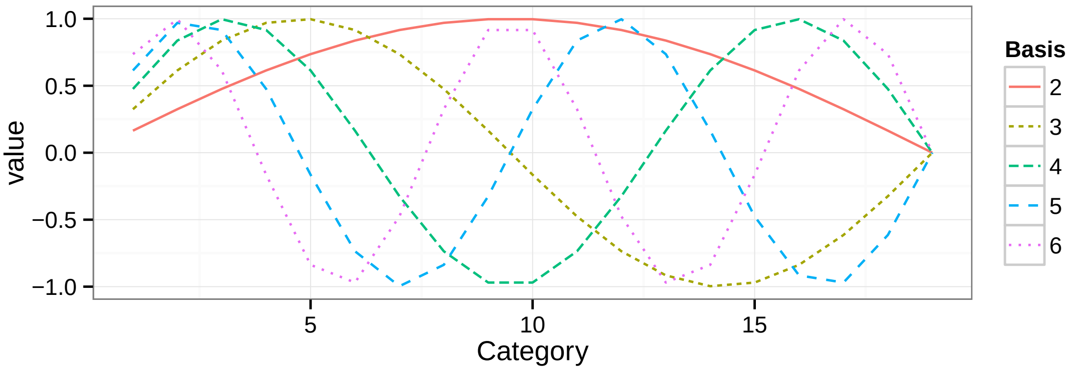

Figure 11. Fourier basis vectors 2 to 6 of a 20 category item. With a large number of categories, it easy to see how the basis vectors approximate a sin wave. The first basis vector is linear (not plotted). The linear basis coefficient represents the mean slope for T_a and the mean position in the latent ability space for T_c. The remaining basis vectors parameterize non-linear variation in spacing. The coefficients of more rapidly oscillating basis vectors (not plotted) can be fixed to 0 to cause the model to ignore high frequency variation.

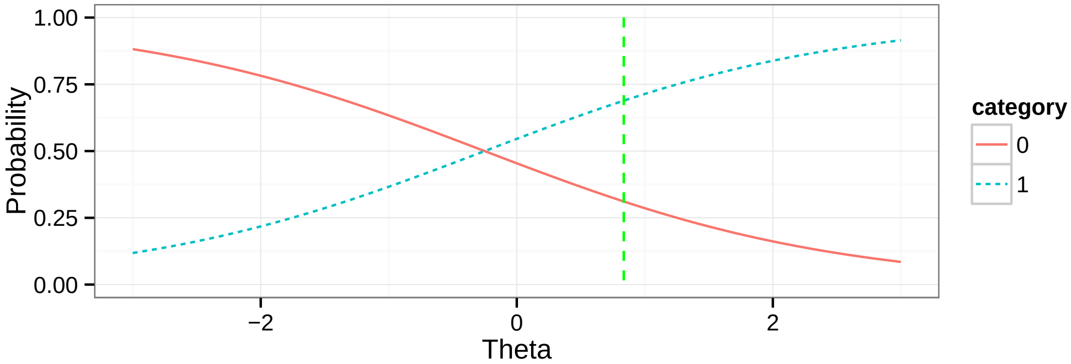

Figure 12. Item characteristic curve for one random 2PL item. Theta of 0.84 is marked with a dashed line. The dashed line intersects category 0 at 0.31 and category 1 at 0.69.

and $\alpha_1 = 1$ (Thissen et al., 2010). To foster an intuitive sense of how this fits together, Figure 10 exhibits some of the simplest Fourier basis models and Figure 11 exhibits the shape of some non-linear basis vectors. Both the popular Partial Credit Model (Masters, 1982) and Generalized Partial Credit Model (Muraki, 1992) can be obtained as parameterizations of the nominal model.

All of the item models presented can be extended to more than 1 dimension or factor by using vectors instead of scalars for the overall slope a and ability θ. For a multidimensional nominal model (Equation 6), the dot product of a and θ is computed first. Thereafter, this overall slope is multiplied by the category specific slope a_k and the result summed with the category specific intercept c_k.

Latent Distribution

A given set of examinee response patterns, item models, and item model parameters imply a latent distribution for the vector of person parameters θ. A common assumption is that the latent distribution is multivariate Normal,

$$L(\theta|\mu, \Sigma) = (2\pi)^{-\frac{k}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\theta - \mu)'\Sigma^{-1}(\theta - \mu)\right]. \tag{8}$$

For example, suppose we have 45 2PL items with random parameters and simulate 1000 response patterns from participants drawn from a Normal distribution with mean 1.5 and standard deviation $\sqrt{1.5}$. To make this example more concrete, let us see how a single response is simulated. The first 2PL item has category response curves as shown in Figure 12. Our first random participant has ability 0.84. Therefore, participant 1 will pick category 0 with probability 0.31 and category 1 with probability 0.69. In this example, category 0 was chosen. The same process is repeated to simulate all response patterns.

Now, suppose we do not know the distribution of θ and want to infer the mean and standard deviation of an assumed multivariate Normal latent distribution from item model parameters and response patterns. Let s be the number of people and x_{ij} be person i's response to item j. The conditional likelihood of person i's response pattern at θ is

$$L(x_i|\theta) = \prod_j \mathrm{P}(\text{pick} = x_{ij}|\theta). \tag{9}$$

Recall that the law of total probability is

$$\mathrm{P}(B) = \sum_v \mathrm{P}(B|A_v)\mathrm{P}(A_v) \tag{10}$$

for event spaces A and B with some partition A_v. One assumption that is required for IFA is *conditional independence*. In this context, conditional independence means that the items are independent after controlling for the latent traits. Using an assumption of conditional independence and the law of total probability, we obtain the unconditional probability of response pattern i,

$$L(x_i) = \int L(x_i|\theta)L(\theta|\mu, \Sigma)\ \mathrm{d}\theta. \tag{11}$$

Recall that Bayes' theorem is

$$\mathrm{P}(A|B) = \frac{\mathrm{P}(B|A)\mathrm{P}(A)}{\mathrm{P}(B)} \tag{12}$$

for event spaces A and B. Since we want to know about the parameters of the latent distribution of ability θ, we need to apply Bayes' theorem to obtain

$$L(\theta|x_i, \mu, \Sigma) = \frac{L(x_i|\theta)L(\theta|\mu, \Sigma)}{L(x_i)}. \tag{13}$$

Thereafter, using an assumption that the response patterns x_i are independently and identically distributed (iid), the moments can be obtained by evaluating

$$\mu^0 = 1 \tag{14}$$

$$\Sigma^0 = \mathrm{diag}(1) \tag{15}$$

$$\mu^{v+1} = \frac{1}{s}\sum_{i=1}^{s} \int \theta L(\theta|x_i, \mu^v, \Sigma^v)\ \mathrm{d}\theta \tag{16}$$

$$\Sigma^{v+1} = \frac{1}{s}\sum_{i=1}^{s} \int \theta\theta^T L(\theta|x_i, \mu^v, \Sigma^v)\ \mathrm{d}\theta - \mu^{v+1}(\mu^{v+1})^T \tag{17}$$

for v iterations until the absolute change in estimates is less than some threshold.

To evaluate these equations, the integrals must be approximated numerically. A variety of numerical integration methods are available such as Gauss-Hermite quadrature, equal interval quadrature, and Monte Carlo integration (Weisstein, n.d.). Monte Carlo integration has been profitably applied to this problem (e.g., Cai, 2010b). Gauss-Hermite quadrature is an attractive choice for 1 dimension, however, the generalization of this integration method to N dimensions raises a bewildering array of potential solutions (e.g., Barajas-Solano, 2012). The current predissertation project will focus on equal interval quadrature because it

	item 1	item 2	item 3	item 4	item 5	item 6	item 7	
$L(x_1 \mid 0.75) =$	**0.32**	0.11	0.48	0.16	0.33	**0.12**	**0.84**	category 0
	0.68	**0.89**	**0.52**	**0.84**	**0.67**	0.88	0.16	category 1

$$L(x_1|0.75) = (0.32)(0.89)(0.52)(0.84)(0.67)(0.12)(0.84) = 0.01$$

Figure 13. Conditional likelihood of response pattern $x_1 = (0, 1, 1, 1, 1, 0, 0)$ for $\theta = 0.75$. Picked categories are highlighted in yellow.

is the simplest method and is still competitive in performance with other leading approaches (Cai, Thissen, & du Toit, 2011).

Integration by equal interval quadrature generalizes to higher dimensions by replicating the same 1 dimensional grid along each dimension. Without loss of generality, let Q be the number of quadrature points per dimension and Q_{width} be the one-sided width of the quadrature for one dimension. Points X_q and areas $A(X_q)$ are arranged as

$$X_q = Q_{width}(1 - \frac{2q}{Q-1}) \quad \text{for } q \in \{0, \ldots, Q-1\} \tag{18}$$

$$A(X_q) = C\ L(X_q|\mu, \Sigma) \tag{19}$$

where C is a normalizing constant to make the areas sum to 1 and $L(\theta|\mu, \Sigma)$ is the multivariate Normal density (Equation 8). It may not be immediately obvious from this definition that the quadrature point locations, X_q, are independent of the latent distribution parameters. To illustrate, an 9 point quadrature of width 3 with latent distribution $\mathcal{N}(1.5, \sqrt{1.5})$ is presented in Table 1. The quadrature points X_q are fixed at the same position regardless of latent distribution parameters μ and Σ. Likewise, note that the conditional probabilities $L(x_i|\theta)$ of Equation 9 do not depend on the latent distribution parameters. These non-dependencies comprise a potential computational advantage of equal interval quadrature over the other methods. Rewriting Equation 11 in terms of quadrature integration obtains

$$L(x_i) = \sum_{q=0}^{Q-1} L(x_i|X_q)A(X_q). \tag{20}$$

To see how this works, it is worthwhile to go through a detailed example. Probabilities conditional on θ are calculated for each response pattern (Figure 13). With the assumption that the latent distribution is multivariate Normal, these conditional probabilities are combined into marginal response probabilities (Table 2). For didactic purposes, $L(x_i|\theta)$ and $L(x_i)$ are described separately. However, it is the proportion $\frac{A(X_q)L(x_i|X_q)}{L(x_i)}$ or $L(X_q|x_i, \mu, \Sigma)$ of Equation 13 that will appear in all subsequent formulas (Table 3).

The moment Equations 16 and 17 can be rewritten in equal interval quadrature inte-

Table 1

Points and areas for a 9 point quadrature of width 3. Note that the areas are not symmetric around zero because the mean of the latent distribution is not 0.

	3	2.25	1.5	0.75	0	-0.75	-1.5	-2.25	-3
Area	0.12	0.22	0.26	0.22	0.12	0.05	0.01	0.00	0.00

Table 2

Example calculation of the unconditional probability of a response pattern. Conditional probabilities $L(x_i|X_q)$ are multiplied by quadrature areas $A(X_q)$ and summed to obtain the marginal response pattern probability, -5.46. All values given in log units.

	3	2.25	1.5	0.75	0	-0.75	-1.5	-2.25	-3
$L(x_i|X_q)$	-10.06	-7.90	-6.06	-4.77	-4.42	-5.36	-7.37	-9.97	-12.87
$A(X_q)$	-2.10	-1.54	-1.35	-1.54	-2.10	-3.04	-4.35	-6.04	-8.10
$A(X_q)L(x_i|X_q)$	-12.15	-9.43	-7.41	-6.30	-6.52	-8.40	-11.72	-16.01	-20.96

Table 3

Example calculation of the moments of a single response pattern.

	3	2.25	1.5	0.75	0	-0.75	-1.5	-2.25	-3
$L(X_q|x_i, \mu, \Sigma)$	0.00	0.02	0.14	0.43	0.35	0.05	0.00	0.00	0.00
1st moment	0.00	0.04	0.21	0.32	0.00	-0.04	-0.00	-0.00	-0.00
2nd moment	0.01	0.10	0.32	0.24	0.00	0.03	0.00	0.00	0.00

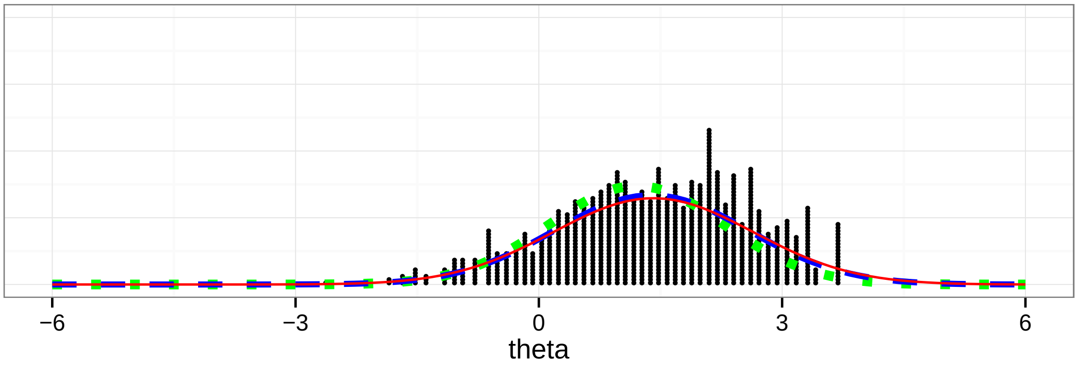

Figure 14. The first 3 estimates of the latent distribution. $\mathcal{N}(0,\ 1)$ was the a priori distribution. The dots show the first estimate, $\mu = 1.2, \sigma = 1.08$. The dashed line shows the second estimate, $\mu = 1.39, \sigma = 1.19$. The solid line shows the third estimate, $\mu = 1.41, \sigma = 1.23$. It takes 9 iterations for estimates to converge to within 10^{-6}, $\mu\ = 1.4, \sigma\ = 1.18$. The y axis is scaled such that the curves have approximately equal area.

gration form

$$\mu^{v+1} = \frac{1}{s}\sum_{j=1}^{s}\frac{1}{L(x_i)}\sum_{q=0}^{Q-1} X_q L(x_i|X_q)A(X_q) \tag{21}$$

$$\Sigma^{v+1} = \frac{1}{s}\sum_{j=1}^{s}\frac{1}{L(x_i)}\sum_{q=0}^{Q-1} X_q X_q^T L(x_i|X_q)A(X_q) - \mu^{v+1}(\mu^{v+1})^T. \tag{22}$$

Iterative estimates of the latent distribution are exhibited in Figure 14. The latent distribution parameters are iteratively estimated as part of the IFA model estimation process without the need to compute derivatives of the latent distribution parameters with respect to the likelihood. Gradients are available for the latent parameters (see Mislevy, 1984), but the gradients are so costly to compute that informal benchmarks suggest that it is faster to estimate the latent parameters iteratively than to employ a gradient descent optimizer.

Multigroup Models

At this point, we can recognize something important about the relationship between items models, item parameters, response patterns, and the latent distribution. By adjusting item parameters in a particular way, the mean and covariance of the latent distribution of θ can be relocated without changing relative response pattern scores. This is the basis of multigroup models. For example, if an identical math test is given to two different classes then we can standardize the latent distribution of one group and free the latent distribution parameters of the other group while simultaneously estimating item parameters. Equality constraints on the item parameters will model the fact that the same test was administered to both groups. If the item parameters are already established (fixed) by prior studies then then latent parameters of both groups can be freed. For single group models, where no reference group is available, the latent distribution is usually fixed to the standard Normal.

To give a sense of the relationship between item parameters and the latent distribution, it will be shown how to relocate the latent distribution by direct adjustment of item parameters. Let μ be a vector of mean adjustments, L be a lower triangle covariance scaling matrix, a be a vector of item slopes, and c a vector of item intercepts. The latent distribution can be moved with

$$a_{adj} = aL \tag{23}$$

$$c_{adj} = c + a\mu. \tag{24}$$

Figure 15 demonstrates a variety of transformations. In all cases, the rank order of response patterns is preserved. Only the x axis is stretched, squeezed, or shifted. Figure 16 exhibits what happens when the mean and covariance get close to the limit of what can be represented within the finite quadrature. The same observed data can be shaped into a range of latent distributions by adjusting item parameters.

Item Parameter Estimation

A major challenge for IFA is efficient estimation of item parameters. Birnbaum (1968) devised the first practical estimation algorithm, Joint Maximum Likelihood (JML). However, JML has three weaknesses. In theory, JML item parameter estimates are inconsistent

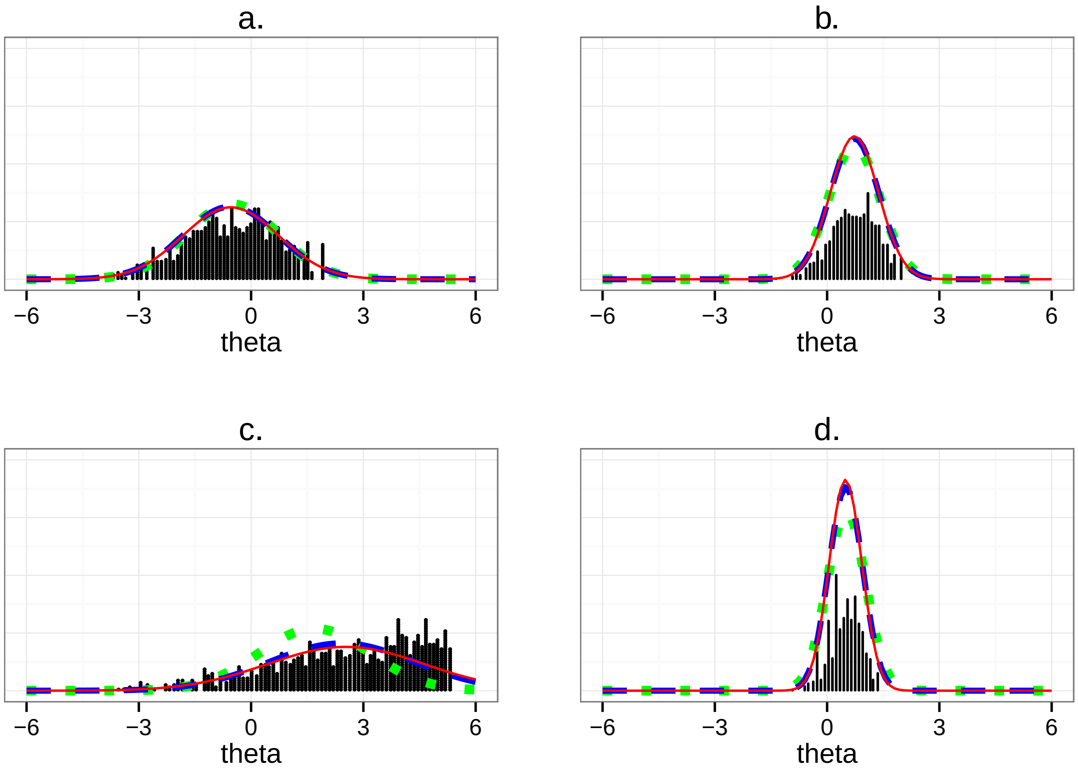

Figure 15. Three iterations of latent distribution parameter estimates for a variety of score distributions. The dots and dashed line show the first and second estimate, respectively. The third estimate, solid line, is estimated at (a) $\mathcal{N}(-0.54,\ 1.28)$, (b) $\mathcal{N}(0.73,\ 0.64)$, (c) $\mathcal{N}(2.53,\ 2.1)$, and (d) $\mathcal{N}(0.49,\ 0.44)$. In all cases, the response patterns and their estimated rank order are identical.

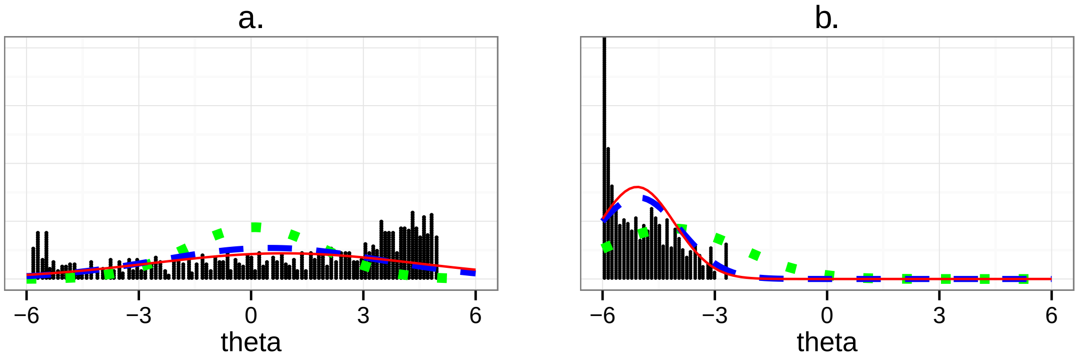

Figure 16. Three iterations of latent distribution parameter estimates for score distributions that are near the limits of the quadrature. The dots, dashed line, and solid line show the first, second, and third estimate, respectively. In plot (a), the variance is too large to be estimated accurately. Many scores are near ceiling or floor. In plot (b), the mean is near -6 which causes many scores to be estimated at floor.

because JML concurrently estimates free parameters for ability, the number of which depend on sample size (Neyman & Scott, 1948). Despite this theoretical weakness, JML performs well enough in practice to attract at least one simulation study of bias (Wang & C.-T. Chen, 2005). It remains unclear how to interpret the bias of an inconsistent estimator. A second, more practical, weakness is that JML does not produce a likelihood statistic suitable for the likelihood ratio test. A third weakness is that latent distribution parameters are not integrated into JML estimation.

Andersen (1972) devised an estimation algorithm known as Conditional Maximum Likelihood (CML) that, in contrast to JML, converges to theoretically consistent item parameter estimates. By design, CML is limited to Rasch models. In Rasch models, all examinees with the same raw score receive the same ability estimate. This is often a workable restriction since the raw score is, in any case, strongly correlated with ability. However, CML involves calculating the sum of products where the number of product terms rapidly increase with the number of items due to the number of possible combinations of obtaining the same sum score. With as few as 100 items, the size of the computation grows to the point that CML cannot run in a reasonable amount of time (Baker & Kim, 2004, p. 135).

Bock and Aitkin (1981) presented the first general item estimation algorithm known as Marginal Maximum Likelihood (MML). MML works with any mix of item models and can estimate more than one latent dimension of ability. MML produces a likelihood statistic suitable for the likelihood ratio test. Other item estimation algorithms have been developed (e.g., Cai, 2010b; Wirth & Edwards, 2007), but MML remains one of the most efficient methods available.

Estimation Software

> An effective algorithm for MML estimation is difficult to program. Effective computer programs must be both computationally and numerically sophisticated (Embretson & Reise, 2000, p. 214)

The best IFA software is commercial (e.g. Cai, Thissen, & du Toit, 2011; Wu, Adams, & Wilson, 1997). However, these are expensive and difficult to use. Furthermore, commercial, black box software cannot be customized. Open-source software is available, but packages suffer from poor software engineering practices. For example, the `mirt` package for `R` is a leading open-source implementation (Chalmers, 2012). However, `mirt` has a monolithic design. Parameter estimation, fit indices, and diagnostics are tightly integrated. It is difficult to take item parameters from another software implementation, such as flexMIRT, and import them into `mirt`. Uncommon skill and effort would be required to combine `mirt` with other types of statistical modeling. `mirt` is designed with the workflow in mind that users input raw ordinal data and receive traditional IFA output.

To facilitate customization and modularity, we have developed a new implementation of MML as a module of the `OpenMx` structural equation modeling (SEM) software (Boker et al., 2011). Researchers who want to combine IFA and SEM in creative ways will not need multiple software packages not designed to work together. In software development, there is always the risk that somebody else will develop similar software first, wasting resources doing the same task twice. The `OpenMx` team releases software as it is developed and invites collaboration as a defense against this problem. For example, the `R` package `rpf` provides mathematics common to IFA estimation algorithms (Pritikin, Weeks, Cai, Houts, & Chalmers, 2013). `rpf` offers the most popular item models and their derivatives. In addition, `rpf` offers a test of conditional independence and Rasch fit statistics. From a software architecture perspective, `rpf` offers modular components of IFA analysis without any item estimation algorithm. I hope to nurture a thriving social community around the project, encouraging other scholars to claim ownership over components and enrich the project with their own vision and contributions.

Marginal Maximum Likelihood

With the unconditional response pattern likelihood P_i, as given in Equation 20, and an assumption that the response patterns x_i are iid, the log likelihood $\mathcal{L}$ of free item and latent distribution parameters ξ is

$$\mathcal{L} = \sum_i \log L(x_i|\xi). \tag{25}$$

However, directly optimizing this likelihood is difficult. The first derivatives with respect to item parameters ξ depend on examinee true scores to obtain per item per response outcome expectations (as in Figure 3). The examinee true scores, in turn, depend on all the free parameters ξ making symbolic expression of the 2nd derivatives very difficult. To obtain relatively simple 2nd derivatives, free parameters ξ are used to compute a provisional estimate of examinee scores. With examinee scores known, observed responses can be partitioned by it. Let w_{joq} be fixed provisional weights for item j at outcome $o \in (0 \dots O)$ at quadrature point X_q. For now, assume that these weights are known. How the weights are obtained will be detailed later. The log likelihood corresponding to the new gradients is

$$\mathcal{L}_{\text{surrogate}} = \sum_j \sum_o \sum_q w_{joq} \log \mathrm{P}(x_{ij} = o | X_q, \xi_j). \tag{26}$$

This likelihood is much easier to optimize for ξ_j than Equation 25 because the items are now independent. The 2nd derivative matrix becomes block diagonal and can be computed analytically. Note that the location of the maximum likelihood estimate is preserved because the new gradients evaluate to 0 in the same location as the original gradients (Bock & Aitkin, 1981). Moreover, when the fixed weights of $\mathcal{L}_{\text{surrogate}}$ are generated with parameters exactly aligned with ξ then $\mathcal{L}$ and $\mathcal{L}_{\text{surrogate}}$ have the same gradient and differ only by a constant. MML consists of v iterations alternating between estimating new weights w_{joq} from ξ^v and optimizing parameters against the weights to find parameter set ξ^{v+1}. The crux of MML is the method of rearranging the conditional likelihood of the response patterns into provisional per-outcome weights. Using initial estimates of item parameters, the weights are estimated as

$$w_{joq} = \sum_i \delta_{(x_{ij}=o)} L(X_q | x_i, \mu, \Sigma) \tag{27}$$

where $\delta_{(x_{ij}=o)}$ is an indicator function that is 1 when $x_{ij} = o$ and 0 otherwise. As the central idea of MML, this transformation deserves a detailed example.

Table 4
Demonstration of how per-outcome weights are obtained. Suppose we have 3 items named u_1, u_2, and u_3 with 2, 2, and 3 possible outcomes, respectively. Some response patterns are in rows and item outcomes are in columns. The weight contribution from person i at $\theta = 0$ is obtained by distributing $L(0|x_i, \mu, \Sigma)$ to the outcome columns according to the response pattern and summing the columns.

	$L(0\|x_i, \mu, \Sigma)$	u_1=0	u_1=1	u_2=0	u_2=1	u_3=0	u_3=1	u_3=2
0,0,0	0.42	0.42	0.00	0.42	0.00	0.42	0.00	0.00
1,0,2	0.53	0.00	0.53	0.53	0.00	0.00	0.00	0.53
1,1,2	0.42	0.00	0.42	0.00	0.42	0.00	0.00	0.42
0,1,0	0.55	0.55	0.00	0.00	0.55	0.55	0.00	0.00
0,1,2	0.56	0.56	0.00	0.00	0.56	0.00	0.00	0.56
1,1,0	0.69	0.00	0.69	0.00	0.69	0.69	0.00	0.00
0,0,1	0.90	0.90	0.00	0.90	0.00	0.00	0.90	0.00
1,0,0	0.58	0.00	0.58	0.58	0.00	0.58	0.00	0.00
total		2.43	2.23	2.44	2.22	2.24	0.90	1.52

Table 4 exhibits the transformation for 3 items and a single quadrature interval. Given that the weights depend on arbitrary starting values that can be very far from maximum likelihood estimates, it may seem surprising that the weights provide enough signal to guide the optimizer toward convergence. However, observe that sum scores are usually positively correlated with true latent scores. In the dichotomous case, for example, the analyst typically knows that category 0 is incorrect and category 1 is correct before any parameter is estimated. Therefore, the orientation of the model is correct even when the difficulty and discrimination parameters may be wildly incorrect. This correct orientation of the model contributes to per-outcome weight accuracy. As the model begins to converge, the weight estimates improve in a virtuous feedback loop. The weights for each quadrature are not summed, but if they were, note that $\sum_q w_{joq}$ is the number of times item j appeared with outcome o across all response patterns,

$$\sum_q w_{joq} = \sum_i \delta_{(x_{ij}=o)} \quad \forall j, o \tag{28}$$

because

$$\sum_q L(X_q|x_i, \mu, \Sigma) = 1 \tag{29}$$

by Equation 20. In other words, the total number of correct and incorrect responses are preserved in Figure 3 despite the partitioning by skill.

Analytic Dimension Reduction

In the factor analysis tradition, IFA dimensions are known as *factors*. As the number of factors increase, the number of quadrature points increase exponentially (Equation 18). To compensate, the number of quadrature points per dimension can be reduced. However, as the number of factors increase beyond 20, another approach is needed. An important optimization is available for the broad class of models that can be restricted to a bifactor or two-tier covariance structure. For example, S. H. Lovibond and P. F. Lovibond (1996) developed a psychological scale to measure depression, anxiety, and stress. Later investigators found that a bifactor model was a good fit to the data, with negative affect as a general factor (Henry & Crawford, 2005, Figure 1). That is, all items load on negative affect to some extent. However, the items are partitioned between depression, anxiety, and stress. In other words, the depression, anxiety, and stress factors are mutually exclusive. Formally, the covariance matrix of a two-tier model is restricted to

$$\Sigma_{\text{two}-\text{tier}} = \begin{pmatrix} G & 0 \\ 0 & \text{diag}(\tau) \end{pmatrix}, \tag{30}$$

where the covariance sub-matrix G is unrestricted (subject to identification), covariance sub-matrix diag(τ) is diagonal, and τ is a vector of variances. The factors that make up G are called primary factors and the factors that comprise τ are called specific factors. Furthermore, each item is permitted to load on at most one specific factor. Continuing our earlier example, stress is a primary factor and anxiety and depression are specific factors.

Let there be p primary dimensions and S specific dimensions. Denote primary latent factors as $\eta = (\eta_1, \ldots, \eta_p)'$ (with covariance sub-matrix G) and specific latent factors as $\xi = (\xi_1, \ldots, \xi_S)'$ (with variances $\tau_1, \ldots, \tau_S$ and no covariance). As in Equation 30, the components of η can be correlated but η and ξ are orthogonal and each specific factor of ξ is orthogonal with all other specific factors. Let $L(\eta, \xi|\mu, \Sigma_{\text{two}-\text{tier}})$ be the multivariate Normal density (Equation 8). Due to orthogonality, the joint distribution of latent factors can be split into

$$L(\eta, \xi|\mu, \Sigma_{\text{two}-\text{tier}}) = \prod_{s=1}^{S} L(\eta|\mu, G) L(\xi_s|\mu, \tau_s). \tag{31}$$

Recall from Equation 9 that the conditional probability of response pattern x_i from participant i for items j is

$$L(x_i|\eta, \xi) = \prod_j \text{P}(\text{pick} = x_{ij}|\eta, \xi). \tag{32}$$

Recall that each item loads on at most one specific factor (Equation 30) and the specific factors are independent (Equations 31). Therefore, Equation 32 is equivalent to

$$L(x_i|\eta, \xi) = \prod_{s=1}^{S} \prod_{j \in \mathcal{I}_s} L(x_{ij}|\eta, \xi_s) \tag{33}$$

where $\mathcal{I}_s$ is a set of items that load on specific factor s. Items that do not load on a specific factor may be grouped into the first specific factor. Using an assumption of conditional independence, the marginal likelihood of the observed data x_i can be written as

$$\begin{aligned} L(x_i) &= \int_{\mathbb{R}^p} \int_{\mathbb{R}^S} L(x_i|\eta, \xi) L(\eta, \xi|\mu, \Sigma) \text{ d}\xi \text{ d}\eta \qquad (34) \\ &= \int_{\mathbb{R}^p} \prod_{s=1}^{S} \int_{\mathbb{R}} \left[\prod_{j \in \mathcal{I}_s} L(x_{ij}|\eta, \xi_s) L(\xi_s|\mu, \tau_s) \right] \text{d}\xi_s \; L(\eta|\mu, G) \text{ d}\eta. \qquad (35) \end{aligned}$$

This algebraic transformation may be easier to follow with just 2 specific factors,

$$\begin{aligned}
&\int_{\mathbb{R}^S} L(x_i|\eta,\xi)L(\xi|\mu,\tau)\ \mathrm{d}\xi \\
&= \int_{\mathbb{R}^S} L(x_i|\eta,\xi_1)L(\xi_1|\mu_1,\tau_1)\ L(x_i|\eta,\xi_2)L(\xi_2|\mu_2,\tau_2)\ \mathrm{d}\xi \\
&= \int_{\mathbb{R}} L(x_i|\eta,\xi_1)L(\xi_1|\mu_1,\tau_1)\ \mathrm{d}\xi_1 \int_{\mathbb{R}} L(x_i|\eta,\xi_2)L(\xi_2|\mu_2,\tau_2)\ \mathrm{d}\xi_2 \\
&= \prod_{s=1}^{2} \int_{\mathbb{R}} L(x_i|\eta,\xi_s)L(\xi_s|\mu_s,\tau_s)\ \mathrm{d}\xi_s.
\end{aligned}$$

If there are many specific factors then this dimension reduction technique is a huge efficiency gain because the $p+S$ dimensional integral is analytically reduced to a product of S $(p+1)$ dimensional integrals.

Rewriting Equation 35 in terms of quadrature integration obtains

$$L(x_i) \approx \underbrace{\sum_{q_p=1}^{Q} \cdots \sum_{q_1=1}^{Q}}_{p\text{-fold}} \prod_{s=1}^{S} \left[\sum_{q=1}^{Q} \prod_{j\in\mathcal{I}_s} L(x_{ij}|(X_{q_1},\ldots,X_{q_p})', X_q) W_q \right] W_{q_1}\ldots W_{q_p} \tag{36}$$

where X_q and W_q are the quadrature points and weights, respectively. Equation 36 is the two-tier equivalent of $L(x_i)$ (Equation 11). To present the two-tier equivalent of $L(\theta|x_i,\mu,\Sigma)$ (Equation 9), we define

$$L(x_i, s|X_{q_1},\ldots,X_{q_p},X_q) = \prod_{j\in\mathcal{I}_s} L(x_{ij}|(X_{q_1},\ldots,X_{q_p})', X_q) \tag{37}$$

$$L(x_i, s|X_{q_1},\ldots,X_{q_p}) = \sum_{q=1}^{Q} L(x_i, s|X_{q_1},\ldots,X_{q_p},X_q)W_q \tag{38}$$

$$L(x_i|X_{q_1},\ldots,X_{q_p}) = W_{q_1}\ldots W_{q_p} \prod_{s=1}^{S} L(x_i, s|X_{q_1},\ldots,X_{q_p}). \tag{39}$$

Thereby, the middle terms are named such that Equation 36 can be rewritten as

$$L(x_i) \approx \underbrace{\sum_{q_p=1}^{Q} \cdots \sum_{q_1=1}^{Q}}_{p\text{-fold}} L(x_i|X_{q_1},\ldots,X_{q_p}). \tag{40}$$

With a few applications of Bayes' theorem (Cai, 2010a), we obtain the conditional contribution of item j for specific dimension s at $(X_{q_1},\ldots,X_{q_p},X_q)$,

$$L(X_{q_1},\ldots,X_{q_p},X_q|x_i,s,\mu,\Sigma) = \frac{L(x_i|X_{q_1},\ldots,X_{q_p})}{L(x_i,s|X_{q_1},\ldots,X_{q_p})} \frac{L(x_i,s|X_{q_1},\ldots,X_{q_p},X_q)}{L(x_i)}. \tag{41}$$

The quantity $L(X_{q_1},\ldots,X_{q_p},X_q|x_i,s,\mu,\Sigma)$ is the two-tier analogue of $L(X_q|x_i,\mu,\Sigma)$ (Equation 13). All equations so far presented work identically by substituting the two-tier analogue in place of $L(X_q|x_i,\mu,\Sigma)$. The moments of the latent distribution (Equations 16 and

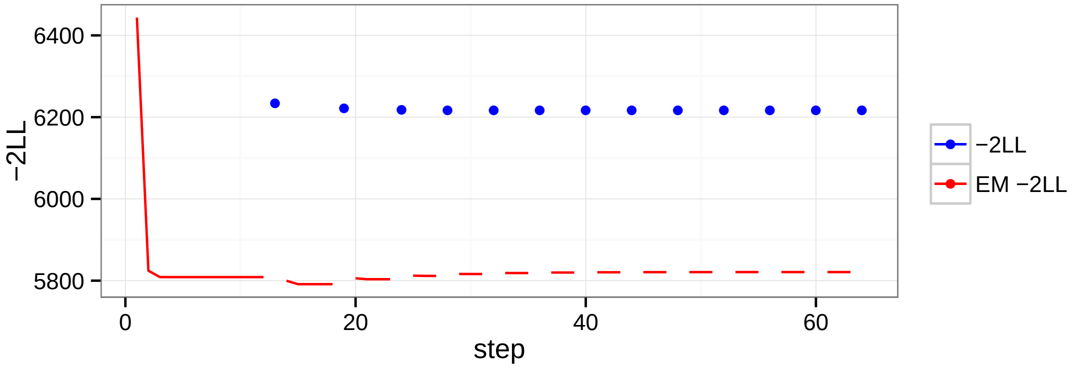

Figure 17. Trajectories of the -2LL and EM -2LL while fitting an example model. The surrogate EM -2LL is monotonically decreasing only within an EM cycle. Estimation is considered to have converged when the maximum absolute parameter change between EM cycles is below some threshold (e.g. 10^{-4}).

17), provisional per-outcome weights (Equations 27 and 28), and the surrogate likelihood of the EM model (Equation 26) all are obtained in exactly the same way whether the we use Equation 13 or 41.

Estimation

In one EM cycle of MML estimation, weights w_{joq} are esimated, the surrogate likelihood is optimized, latent distribution parameters are updated based on fresh estimates, and iteration is terminated if the maximum absolute change in free parameters is less than some threshold (e.g. 10^{-4}). See Figure 17 for an example of the progression of the two likelihoods during model estimation. Since analytic 1st and 2nd derivatives are available for the 4PL model, graded response model, and nominal model (Baker & Kim, 2004; Chalmers, 2012), the surrogate likelihood can be optimized using the Newton-Raphson procedure (Luenberger & Ye, 2008, Section 8.8). Let p_n be the parameter vector after n cycles, new estimates are obtained by

$$p_{n+1} = p_n - f''(p_n)^{-1} f'(p_n). \tag{42}$$

Inverting the Hessian $f''(p_n)$, a square matrix of second-order partial derivatives with respect to the likelihood, is often a performance bottleneck for models with more than a few hundred parameters. Fortunately, IFA items are assumed independent from each other after controlling for the latent factors. Since items are independent, their derivatives are as well, resulting in a block diagonal Hessian. The inverse of a block diagonal matrix consists of the inverse of the blocks. Equality constraints are accommodated by counting the number of times that a parameter is used and entry-wise dividing the inverse Hessian by the outer product of $(e_1, \ldots, e_k)$ where e_k is the number of times that the parameter k appears in an item model (Petersen & Pedersen, 2006, Derivatives of an Inverse).

Optimization does not always go as smoothly as might be surmised from Figure 17. For instance, one data set with more than 50% missingness in half of the items is more

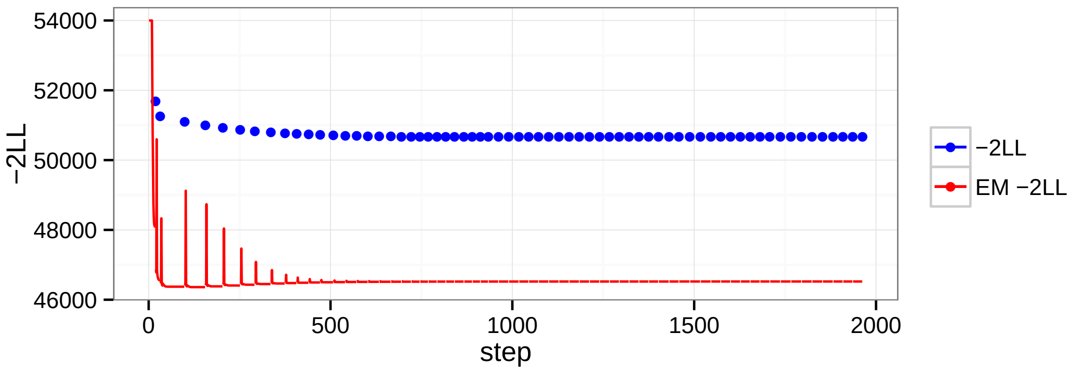

Figure 18. Trajectories of the -2LL and EM -2LL while fitting a model with 50% missing data in half of the items. The initial spikes appear because the damping algorithm needs to see three parameter updates before evaluating whether overshoot is getting worse (Ramsay, 1975). The spikes at the beginning of the EM cycles are due to parameters going so far away from reasonable values that the optimizer restarts from initial values with more damping. By default, the optimizer optimistically trusts the derivatives and does not evaluate the surrogate EM -2LL. Once a restart occurs, the optimizer starts watching the surrogate EM -2LL and finishes early if the -2LL starts rising too far above the best-so-far -2LL. In this plot, the surrogate EM -2LL values are clipped at 54000 because the surrogate EM -2LL peak reaches as high as 742305.53. Without clipping, the small changes in the likelihoods are barely visible.

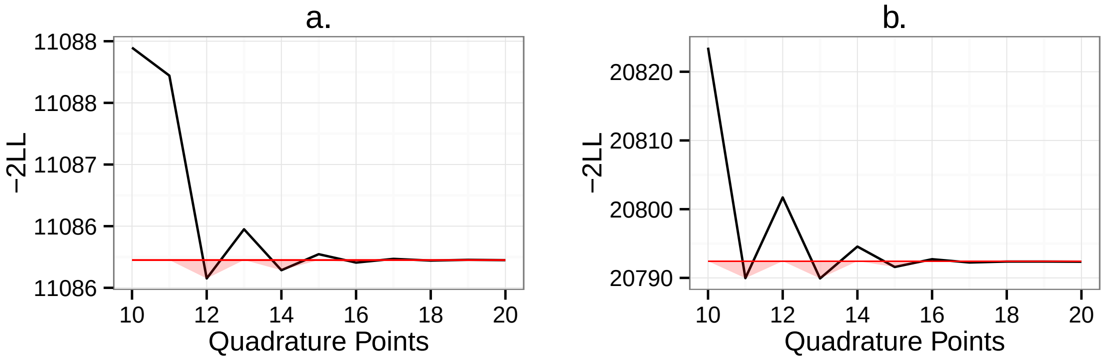

Figure 19. Model fit as a function of quadrature points. In plot (a), the fit of a single group model is estimated a fraction of a point better than the true fit when 12 quadrature points are used for integration. In plot (b), a two group model, the better than true fits occur with 11 and 13 quadrature points. In both plots, the shading highlights models estimated with a better than true fit.

stubborn. The fitting process of this model is exhibited in Figure 18. Early on, the Newton steps tend to overshoot, oscillating with ever larger magnitude. To compensate, a damping algorithm is used when this behavior is detected (Ramsay, 1975). A number of such ad hoc heuristics are necessary to help the Newton-Raphson method approach its theoretical efficiency (Luenberger & Ye, 2008, Section 8.8).

The shape of the quadrature has an influence on model fit. As already discussed, the width of the quadrature places a limit on the range of means and variances that can be estimated (review Figure 16). In addition, increasing the number of quadrature points is the main way to gain accuracy at the cost of increased estimation time. Estimation time is typically dominated by the number of points raised to the number of factors (raised to the number of primary factors +1 in two-tier models) (Cai, 2010b). For high dimensional models, it is of practical importance to minimize the number of points. However, as the number of quadrature points is reduced, there is risk of obtaining a better than true fit (Figure 19). Hence, high dimensional models are not feasible to estimate because the number of quadrature points required for accurate estimation is too large to evaluate in a reasonable amount of time.

Scoring

Once a model is fit, there are a number of ways to obtain scores for individual responses. All methods use either maximum likelihood (ML) or integration over the latent distribution. ML methods find the modal estimate while methods that integrate over the latent distribution obtain the mean estimate. ML methods may have applications in special circumstances, but integrating over the latent distribution is more robust. The 3PL item model can have a bimodal likelihood surface (Thissen & Orlando, 2001, p. 136) that will cause ML estimate to be sensitive to starting values. For details of ML scoring methods see Embretson and Reise (2000).

One method is Expected a Posteriori (EAP; Bock & Mislevy, 1982; Cai, 2010a). For response pattern i, EAP scores are computed as

$$\hat{\mu}_i = \frac{1}{L(x_i)} \int \theta L(x_i|\theta) L(\theta|\mu, \Sigma)\ \mathrm{d}\theta \tag{43}$$

$$\hat{\Sigma}_i = \left[\frac{1}{L(x_i)} \int \theta\theta^T L(x_i|\theta) L(\theta|\mu, \Sigma)\ \mathrm{d}\theta \right] - \hat{\mu}_i \hat{\mu}_i^T . \tag{44}$$

Note the close similarity between EAP Equations 43 and 44 and latent distribution Equations 16 and 17. This is one of the advantages of EAP scores over other scoring methods. They are consistent with estimates of latent distribution parameters.

Simulation Studies of Bias

In order to test the accuracy of our new `OpenMx` MML implementation, two simulation studies were conducted. Study 1 examines bias under different conditions of missing data. Since MML is a full information estimator, missing data should not introduce bias. Study 2 replicates Simulation 1 from Cai, Yang, and Hansen (2011) to demonstrate that the two-tier analytic dimension reduction technique is working correctly.

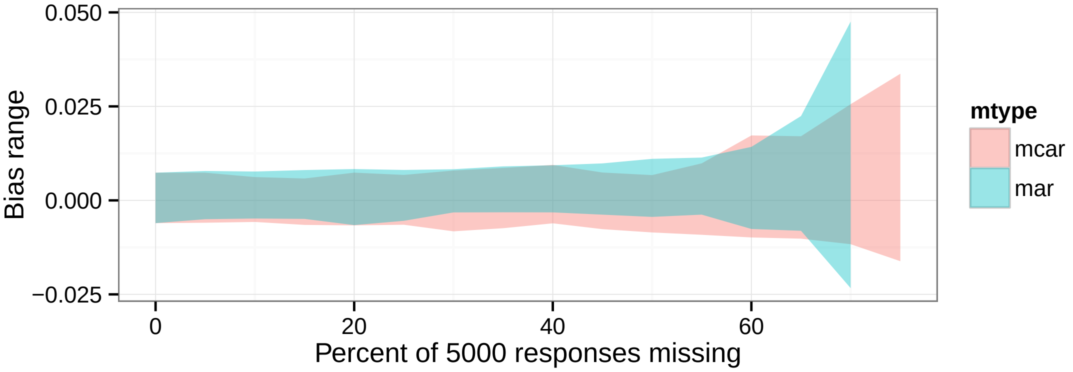

Figure 20. Bias range plotted against percent missing by two missingness conditions. The two missingness conditions are missing completely at random (MCAR) and missing at random (MAR). The plots stop around 70% because item parameters cannot be estimated when all data is missing in a columns.

Study 1. Newton-Raphson convergence criterion was set at 10^{-7} and the EM cycles were considered converged when the absolute difference in -2LL was less than 10^{-4}. For the missing data study, the quadrature was set to 31 points evenly spaced between -5 and 5. Generating parameters are exhibited in Tables 5 and 6. The full data consisted of 5000 simulated response patterns. The two missing data conditions were missing completely at random (MCAR) and missing at random (MAR). MAR is a type of missingness where data is missing as a function of observed data. The MAR condition was operationalized by randomly erasing data in the last 15 items prioritizing by the sum score of the first 5 items. Let $\hat{b}_m$ be the estimated parameter set on the mth Monte Carlo replication. For M replications, the Monte Carlo bias is defined as

$$\left[\frac{1}{M}\sum_{m}^{M}\hat{b}_m\right] - b.$$

Five hundred replications were conducted. Complete source code is available in Appendix A. Results by percent missing and missingness condition are exhibited in Figure 20. The maximum absolute bias was 0.01 across both missingness conditions from 0 to 60% missing.

Table 5
Generating 2PL item parameters i1-i10 for simulation study of bias with two conditions of missingness.

	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10
slope	0.73	0.66	1.18	1.28	1.33	2.13	0.73	1.75	0.99	1.51
intercept	0.18	1.60	-0.82	0.74	-0.31	0.39	-2.21	-0.04	0.94	0.59

Study 2. Closely following the simulation study Cai, Yang, and Hansen (2011, Simulation 1), Newton-Raphson convergence criterion was set at 10^{-7} and EM cycles were considered converged when the absolute difference in -2LL was less than 10^{-4}. The quadrature was set to 21 points evenly spaced between -5 and 5. Generating parameters are

Table 6
Generating 2PL item parameters i11-i20 for simulation study of bias with two conditions of missingness.

	i11	i12	i13	i14	i15	i16	i17	i18	i19	i20
slope	1.58	1.04	1.36	0.93	0.79	1.97	1.21	0.50	0.82	1.73
intercept	0.78	-1.99	-0.06	-1.47	0.42	-0.10	-0.05	-0.41	-0.06	0.76

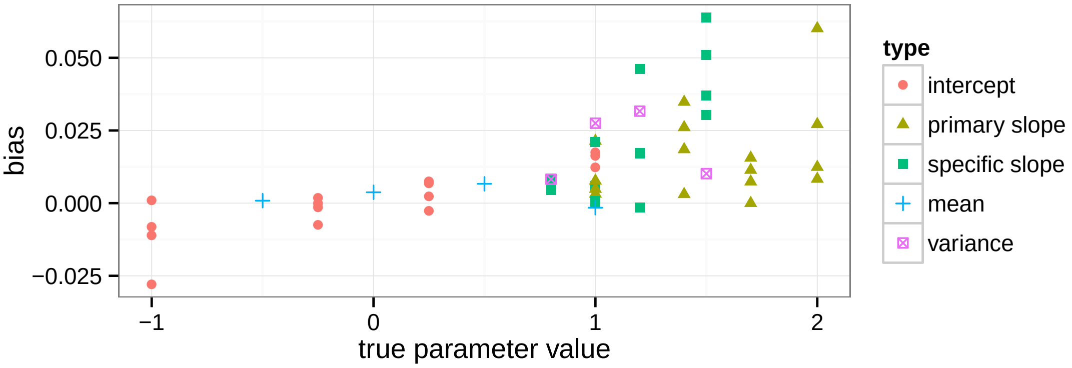

Figure 21. Bias by true parameter value and parameter type. Maximum absolute bias was 0.06 after 500 replications. A similar pattern of bias was evident in Cai, Yang, and Hansen (2011).

exhibited in Table 7. One thousand response patterns were simulated for each group. Five hundred replications were conducted. Complete source code is available in Appendix B. The biases of primary slopes, specific slopes, intercepts, means, and variances (Figure 21) were indistinguishable from those of Cai, Yang, and Hansen (2011). A maximum absolute bias of 0.06 was obtained demonstrating acceptably low bias.

Discussion

IFA lets us place items and persons on the same scale, compare the ability distributions of groups, analyze item bias, and construct optimal tests for specific applications (Mislevy, 1993). For example, IFA is used by The College Board, Educational Testing Service, Law School Admission Council, and Association of American Medical Colleges to evolve standardized tests while maintaining a stable measurement scale. Tests such as the SAT, GRE, LSAT, and MCAT use primarily dichotomous scoring, but IFA models are also well suited to many types of psychological assessment instruments.

Item parameter estimation is the challenging part of IFA analysis. MML is one of the leading algorithms. Commercial software for MML is available, but it is Windows platform-centric and customization possibilities are limited. A high quality open-source implementation of MML is important to invite more researchers to take advantage of IFA and foster innovation.

Our MML implementation for `OpenMx` exhibits acceptably low bias in the presence of missing data (Study 1) and efficiently recovers parameters from simulated data in a two-tier model (Study 2). Anecdotal evidences suggests that estimation time is comparable with

Table 7

Generating parameters for Cai, Yang, and Hansen (2011, Simulation 1), a two group bifactor model. The means and variances of Group 1 are fixed to standard Normal. The generating parameters for the means and variances of Group 2 are $\mu_1 = 1, \sigma_1^2 = 0.8, \mu_2 = -0.5, \sigma_2^2 = 1.2, \mu_3 = 0, \sigma_3^2 = 1.5$, and $\mu_4 = 0.5, \sigma_4^2 = 1$. Items 13-16 are only present in Group 1. Items present in both groups had their parameters constrained equal.

Item	Primary Slope	Specific Factor	Specific Slope	Intercept
1	1.00	2	0.80	1.00
2	1.40	2	1.50	0.25
3	1.70	2	1.20	-0.25
4	2.00	2	1.00	-1.00
5	1.40	3	1.00	1.00
6	1.70	3	0.80	0.25
7	2.00	3	1.50	-0.25
8	1.00	3	1.20	-1.00
9	1.70	4	1.20	1.00
10	2.00	4	1.00	0.25
11	1.00	4	0.80	-0.25
12	1.40	4	1.50	-1.00
13	2.00	5	1.50	1.00
14	1.00	5	1.20	0.25
15	1.40	5	1.00	-0.25
16	1.70	5	0.80	-1.00

flexMIRT, a leading commercial software implementation (Cai, 2012). For one randomly selected repetition of Study 2, OpenMx fit the model in 9.5s while flexMIRT took 13s. However, flexMIRT started with worse starting values that cost 2-3 EM cycles and spent an extra 25 EM cycles (out of 211 total EM cycles) to improve the -2LL by 0.0017 beyond what OpenMx achieved. Overall, flexMIRT EM cycles appear slightly more efficient than OpenMx because flexMIRT tends to achieve the same -2LL with fewer EM cycles. We are keenly interested to learn the cause of these differences, but they are probably of little practical importance. In most models we have compared against flexMIRT, the -2LL matches to the hundredth's place. That is probably good enough for routine statistical modeling in psychology.

OpenMx can estimate models with 1000 items. Simultaneous estimation of 2000 items is probably feasible but has not been attempted. The number of items per model is constrained by Equation 9. As the number of items increase, the product in Equation 9 becomes smaller than can be represented with double-precision floating point, roughly 10^{-308}. Since probabilities are always less than 1, OpenMx extends the range to roughly 10^{-616}. It is not clear how to represent smaller numbers without using arbitrary precision arithmetic or converting to a logarithmic representation. Since the evaluation of Equation 9 is a performance bottleneck for MML, use of a numeric representation that is not hardware optimized would have noticeably adverse consequences on performance.

Data size is limited to fit in random access memory by R. Internally, all calculations are organized to operate on the unit of a single response pattern. Therefore, it should be possible to relax this constraint and allow data larger than can fit in random access memory. Estimation time should increase no faster than linearly with the number of response patterns.

Future Directions

There are two conspicuous omissions from this report. I have not yet implemented estimation of standard errors (SEs) for item parameters. SEs are early on my list of features to add. The other crucial missing piece is the option of Bayesian priors on item parameters. Bayesian priors are usually required to estimate the upper and pseudo-guessing bound of the 4PL model (Equation 3).

A list of further possible improvements is lengthy. IFA analysis makes two assumptions: multivariate Normality of the latent scores and conditional independence. It would be important to be able to relax the assumption of Normality using splines (Woods, 2006) or empirical histograms (Woods, 2007). It would also be important to offer a test for conditional independence (e.g., W.-H. Chen & Thissen, 1997).

EAP scores are compelling because they are consistent with estimates of latent distribution parameters. However, one characteristic of EAP scores may be undesirable in some contexts. When Rasch constraints are not used, the same sum score can obtain different EAP scores. For example, response pattern 001 may obtain a higher score than 010 even though both patterns have a sum score of 1 if the last item is more discriminating than the 2nd item. The sum-score EAP procedure offers a way to estimate sum scores for non-Rasch models (Thissen, Pommerich, Billeaud, & Williams, 1995) and would make a useful addition. It would also be nice to offer maximum likelihood based scoring algorithms, if only to compare against other IFA software. The derivatives of item parameters with respect to skill are available from `rpf`. These are used for computing the standard error of response pattern scores (a.k.a., the standard error of estimate) and are the main ingredient for maximum likelihood based scoring algorithms.

MML as an `OpenMx` module opens up possibilities for combining IFA and SEM approaches. We hope to combine IFA with standard behavioral genetics, decomposing the observed variance into additive genetics, common environments, and unique environments components; use random coefficients for the IFA parameters such that not everybody has the same item-factor relationship; and add the ability to include an IFA measurement model as part of a longitudinal SEM (e.g., modeling manifests for a single factor with IFA and modeling the latent factors with a SEM growth curve model).

Conclusions

Tests, questionnaires, and similar instruments produce a particular kind of data that is best analyzed with IFA. `OpenMx`, a freely available open-source statistical software, is now capable of estimating IFA models. However, datasets often include other indicators measured on an interval or ratio scale. For example, in an educational context, students are often recorded with an age and a socioeconomic score. Many workers want to understand how school performance is influenced by demographic factors in a general SEM setting. There is some work in this direction (e.g., Adams, Wilson, & Wang, 1997). However, at the time of writing, there is still no consensus on how to best combine IFA and SEM approaches without resorting to Monte Carlo methods (Cai, 2013). The availability of an IFA algorithm in capable, open-source SEM software is a step toward the unification of IFA and SEM.

During the development of `OpenMx` IFA software, some of the potential advantages of open-source software (Free Software Foundation, n.d.) have already borne fruit. Phil Chalmers, maintainer of `mirt`, has closely followed `OpenMx` change sets. When we added a damping algorithm to the Newton-Raphson optimizer, Chalmers added the same algorithm to accelerate EM convergence in `mirt`. A fit statistic known as *standardized outfit* (Wright & Masters, 1982) was computed incorrectly by `mirt` and other open-source IFA software. We recognized the problem and Chalmers put together a fix. Until recently, `mirt` was limited to bifactor models and did not support two-tier models (Cai, 2010a). We recognized how to algebraically reorganize the computation, facilitating the two-tier optimization without excessive duplication of code. In an effort to speed estimation performance, we recognized that computing Equation 9 in log space was counterproductive. Subsequently, Chalmers removed the log from the corresponding code in `mirt`. We pointed out a bug in `mirt`'s computation of the local dependence χ^2 statistic (W.-H. Chen & Thissen, 1997) and in the calculation of the outcome probabilities for the nominal response model at large positive θ. These bugs were subsequently fixed.

Our contributions from `OpenMx` to `mirt` are highlighted, but the collaboration has certainly flowed in both directions. Foremost, `OpenMx` uses the item model derivatives from `mirt`. The derivation of these derivatives is documented (Baker & Kim, 2004), but it still would require a great deal of effort to derive and validate a software implementation. Furthermore, the estimation of group latent distribution parameters was greatly clarified by study of `mirt`'s source code.

In the adaption of `OpenMx` for IFA, we added a Newton-Raphson optimizer and *Compute objects*, a small domain-specific language for controlling the `OpenMx` engine. These additions are of general utility. For example, the Newton-Raphson optimizer could be used to optimize SEM restricted to the case where first and second derivatives are available and there are no non-linear constraints. This could help performance because the Newton-Raphson method is one of the most efficient optimization methods available (Luenberger & Ye, 2008). Compute objects provide a way to customize `OpenMx` that is less difficult than modifying the source code but more powerful than hitherto available. For example, no other IFA packages that we are aware of permit easy customization of the EM loop. It is not clear how this feature will be exploited, but it puts more power into the hands of the user.

Much work remains, but I have already begun to use `OpenMx` for fitting IFA models in my own research. Creative use of equality constraints combined with the likelihood ratio test can already yield a wide array of results. I am confident that the convenience of working with IFA models in `OpenMx` will be a popular choice with other statisticians.

References

Adams, R. J., Wilson, M., & Wang, W.-c. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement*, *21*(1), 1–23. (Cit. on p. 28).

Andersen, E. B. (1972). The numerical solution of a set of conditional estimation equations. *Journal of the Royal Statistical Society, Series B*, *34*, 42–54. (Cit. on p. 17).

Baker, F. B. & Kim, S. H. (2004). *Item response theory: Parameter estimation techniques* (2nd). CRC Press. (Cit. on pp. 17, 22, 29).

Barajas-Solano, D. (2012, January). On sparse-grid quadratures for multiple integrals with Gaussian weight [Web log post]. Retrieved October 14, 2013, from http://dbarajassolano.wordpress.com/2012/01/26/on-sparse-grid-quadratures/. (Cit. on p. 12)

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick (Eds.), *Statistical theories of mental test scores* (pp. 397–479). Reading, MA: Addison-Wesley. (Cit. on p. 15).

Bock, R. D. & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika*, *46*, 443–459. (Cit. on pp. 17, 18).

Bock, R. D. & Mislevy, R. J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement*, *6*(4), 431–444. (Cit. on p. 24).

Boker, S. M., Neale, M., Maes, H., Wilde, M., Spiegel, M., Brick, T., ... Bates, T., et al. (2011). OpenMx: An open source extended structural equation modeling framework. *Psychometrika*, *76*(2), 306–317. (Cit. on p. 17).

Cai, L. (2010a). A two-tier full-information item factor analysis model with applications. *Psychometrika*, *75*, 581–612. doi:10.1007/s11336-010-9178-0. (Cit. on pp. 21, 24, 29)

Cai, L. (2010b). High-dimensional exploratory item factor analysis by a Metropolis-Hastings Robbins-Monro algorithm. *Psychometrika*, *75*, 33–57. (Cit. on pp. 12, 17, 24).

Cai, L. (2012). flexMIRT: A numerical engine for multilevel item factor analysis and test scoring [Computer software]. Version 1.88. Vector Psychometric Group. (Cit. on p. 27).

Cai, L. (2013, October). *Flexible multidimensional item analysis, test scoring, and model fit evaluation*. Talk presented at the annual conference of the Society of Multivariate Experimental Psychology, St. Petersburg, Florida. (Cit. on p. 29).

Cai, L., Thissen, D., & du Toit, S. (2011). IRTPRO [Software manual]. Version 2.1. Scientific Software International. (Cit. on pp. 13, 17).

Cai, L., Yang, J. S., & Hansen, M. (2011). Generalized Full-Information Item Bifactor Analysis. *Psychological Methods*, *16*(3), 221–248. (Cit. on pp. 24–27).

Camilli, G. (1994). Teacher's corner: Origin of the scaling constant d= 1.7 in Item Response Theory. *Journal of Educational and Behavioral Statistics*, *19*(3), 293–295. (Cit. on p. 6).

Chalmers, R. P. (2012). mirt: a multidimensional item response theory package for the R environment. *Journal of Statistical Software*, *48*(6), 1–29. Retrieved from http://www.jstatsoft.org/v48/i06/. (Cit. on pp. 17, 22)

Chen, W.-H. & Thissen, D. (1997). Local dependence indexes for item pairs using Item Response Theory. *Journal of Educational and Behavioral Statistics*, *22*(3), 265–289. (Cit. on pp. 28, 29).

Cronbach, L. & Furby, L. (1970). How should we measure change – Or should we? *Psychological Bulletin*, *74*, 68–80. (Cit. on p. 7).

Embretson, S. E. & Reise, S. P. (2000). *Item Response Theory for psychologists*. Lawrence Erlbaum. (Cit. on pp. 6, 17, 24).

Free Software Foundation. (n.d.). Free Software Movement. Retrieved from http://www.gnu.org/philosophy/free-software-intro.html. (Cit. on p. 29)

Henry, J. D. & Crawford, J. R. (2005). The short-form version of the Depression Anxiety Stress Scales (DASS-21): Construct validity and normative data in a large non-clinical sample. *British Journal of Clinical Psychology*, *44*(2), 227–239. (Cit. on p. 20).

Loken, E. & Rulison, K. L. (2010). Estimation of a four-parameter item response theory model. *British Journal of Mathematical and Statistical Psychology*, *63*(3), 509–525. (Cit. on p. 7).

Lord, F. M., Novick, M. R., & Birnbaum, A. (1968). Statistical theories of mental test scores. (Cit. on pp. 4, 6).

Lovibond, S. H. & Lovibond, P. F. (1996). *Manual for the Depression Anxiety Stress Scales.* Psychology Foundation of Australia. (Cit. on p. 20).

Luenberger, D. G. & Ye, Y. (2008). *Linear and nonlinear programming.* Springer. (Cit. on pp. 22, 24, 29).

Masters, G. N. (1982). A Rasch model for partial credit scoring. *Psychometrika*, *47*, 149–174. (Cit. on p. 11).

Mislevy, R. J. (1984). Estimating latent distributions. *Psychometrika*, *49*(3), 359–381. (Cit. on p. 15).

Mislevy, R. J. (1993). Foundations of a new test theory. In N. Frederiksen, R. J. Mislevy, & I. I. Bejar (Eds.), *Test theory for a new generation of tests* (pp. 19–39). New Jersey: Lawrence Erlbaum Hillsdale. (Cit. on p. 26).

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, *16*, 159–176. (Cit. on p. 11).

Neyman, J. & Scott, E. L. (1948). Consistent estimates based on partially consistent observations. *Econometrica*, *16*, 1–32. (Cit. on p. 17).

Petersen, K. B. & Pedersen, M. S. (2006). The matrix cookbook. Technical University of Denmark. (Cit. on p. 22).

Pritikin, J. N., Weeks, J., Cai, L., Houts, C., & Chalmers, R. P. (2013). *Response Probability Functions* [Computer software]. Version 0.16. Retrieved October 15, 2013, from http://cran.r-project.org/web/packages/rpf/index.html. (Cit. on p. 18)

Ramsay, J. O. (1975). Solving implicit equations in psychometric data analysis. *Psychometrika*, *40*(3), 337–360. (Cit. on pp. 23, 24).

Reckase, M. D. (1985). The difficulty of test items that measure more than one ability. *Applied Psychological Measurement*, *9*(4), 401–412. (Cit. on p. 6).

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement*, *34*(4), 100. (Cit. on p. 7).

Thissen, D., Cai, L., & Bock, R. D. (2010). The Nominal Categories Item Response Model. In M. L. Nering & R. Ostini (Eds.), *Handbook of Polytomous Item Response Theory Models* (pp. 43–75). Routledge. (Cit. on pp. 7, 11).

Thissen, D. & Orlando, M. (2001). IRT for items scored in two categories. In D. Thissen & H. Wainer (Eds.), *Test scoring.* Lawrence Erlbaum Associates, Inc. (Cit. on p. 24).

Thissen, D., Pommerich, M., Billeaud, K., & Williams, V. S. (1995). Item response theory for scores on tests including polytomous items with ordered responses. *Applied Psychological Measurement*, *19*(1), 39–49. (Cit. on p. 28).

Thissen, D. & Steinberg, L. (1986). A taxonomy of item response models. *Psychometrika*, *51*(4), 567–577. (Cit. on p. 7).

Wang, W.-C. & Chen, C.-T. (2005). Item parameter recovery, standard error estimates, and fit statistics of the Winsteps program for the family of Rasch models. *Educational and Psychological Measurement*, *65*(3), 376–404. (Cit. on p. 17).

Weisstein, E. (n.d.). Numerical integration. From MathWorld–A Wolfram Web Resource. Retrieved October 14, 2013, from http://mathworld.wolfram.com/NumericalIntegration.html. (Cit. on p. 12)

Wirth, R. J. & Edwards, M. C. (2007). Item factor analysis: Current approaches and future directions. *Psychological Methods*, *12*(1), 58. (Cit. on p. 17).

Woods, C. M. (2006). Ramsay-curve Item Response Theory (RC-IRT) to detect and correct for nonnormal latent variables. *Psychological methods*, *11*(3), 253. (Cit. on p. 28).

Woods, C. M. (2007). Empirical histograms in Item Response Theory with ordinal data. *Educational and Psychological Measurement*, *67*(1), 73–87. (Cit. on p. 28).

Wright, B. D. & Masters, G. N. (1982). *Rating scale analysis*. Chicago: MESA Press. (Cit. on p. 29).

Wu, M., Adams, R., & Wilson, M. (1997). ConQuest: Multi-aspect test software. Australian Council for Educational Research. (Cit. on p. 17).

Appendix A
Source Code for Study 1

```
#options(error = utils::recover)
library(OpenMx)
library(rpf)

calc.bias <- function (bank, correct) {
  bias <- matrix(0, nrow=dim(correct)[1], ncol=dim(correct)[2])
  for (sx in 1:length(bank)) {
    bias <- bias + bank[[sx]]$param
  }
  bias <- (bias / length(bank)) - correct
  bias
}

mcar <- function(data, pct) {
  size <- prod(dim(data))
  erase <- rep(TRUE, size * pct)
  mask <- c(erase, rep(FALSE, size - length(erase)))[order(runif(size))]
  shaped.mask <- array(mask, dim=dim(data))
  data[shaped.mask] <- NA
  # remove when all items are missing
  data <- data[apply(is.na(data), 1, sum) != dim(data)[2],]
  data
}

mar <- function(data, pct) {
  few <- dim(data)[2] / 4
  data.o <- order(apply(sapply(data[,1:few], unclass), 1, sum))
  size <- prod(dim(data))
  erase <- rep(TRUE, size * pct)
  dep.dim <- c(dim(data)[1], (dim(data)[2] - few))
  depsize <- prod(dep.dim)
  mask <- c(erase, rep(FALSE, depsize - length(erase)))
  mask <- matrix(mask, nrow=dep.dim[1], ncol=dep.dim[2], byrow=TRUE)
  shaped.mask <- array(FALSE, dim=dim(data))
  shaped.mask[,(few+1):dim(data)[2]] <- mask
  shaped.mask <- shaped.mask[data.o,]
  data[shaped.mask] <- NA
  # remove when all items are missing
  data <- data[apply(is.na(data), 1, sum) != dim(data)[2],]
  data
}

set.seed(1)
numItems <- 20
correct <- sapply(1:numItems, function (x) rpf.rparam(rpf.grm()))
# probably should re-run simulation with round(correct,2) parameters TODO

fit1 <- function(param1) {
  result <- list()
  for (c in colnames(param1)) {
    result[[c]] <- param1[[c]]
  }

  items <- list()
  items[1:numItems] <- rpf.grm()

  set.seed(param1$seed)
  fulldata <- rpf.sample(param1$n, items, correct)
  if (param1$mtype == "mar") {
    data <- mar(fulldata, param1$missing)
  } else if (param1$mtype == "mcar") {
    data <- mcar(fulldata, param1$missing)
  }
  for (c in colnames(data)) { attr(data[,c], 'mxFactor') <- TRUE }

  m.mat <- mxMatrix(name="mean", nrow=1, ncol=1, values=0, free=FALSE)
  cov.mat <- mxMatrix(name="cov", nrow=1, ncol=1, values=1, free=FALSE)

  ip.mat <- mxMatrix(name="ItemParam", nrow=2, ncol=numItems,
                     values=c(1,0), free=TRUE)
  m1 <- mxModel(model="drm1", ip.mat, m.mat, cov.mat,
              mxData(observed=data, type="raw"),
              mxExpectationBA81(
                ItemSpec=items,
                ItemParam="ItemParam",
                mean="mean", cov="cov", qwidth=5, qpoints=31),
              mxFitFunctionML(),
                mxComputeSequence(steps=list(
                mxComputeIterate(steps=list(
                  mxComputeOnce('expectation', context='EM'),
                  mxComputeNewtonRaphson(free.set='ItemParam'),
                  mxComputeOnce('expectation'),
                  mxComputeOnce('fitfunction', free.set=c("mean","cov"),
                                maxAbsChange=TRUE)
                )),
                mxComputeOnce('fitfunction', free.set=c("mean","cov"),
                              fit=TRUE))))
  m1 <- mxRun(m1, silent=TRUE)

  result$LL <- m1@fitfunction@result
  result$param <- m1@matrices$ItemParam@values
  result
}

param <- expand.grid(seed=1:500, n=5000, mtype=c('mar','mcar'),
                     missing=seq(0,.9,.05))

rda <- "missing.rda"
if (0) {
  sim <- list()
} else {
  load(rda)
  sim <- missing
}
if (1) {
  print(paste("Estimating",dim(param)[1], "simulations"))
  for (cnt in 1:dim(param)[1]) {
    p1 <- param[cnt,]
    got <- sapply(sim, function(s) s$seed == p1$seed &
                  s$mtype == p1$mtype &
                  s$missing == p1$missing & s$n == p1$n)
    if (any(got)) {
      print(paste("found",cnt))
      next
    }
    #print(param[cnt,])
    sim[[length(sim)+1]] <- fit1(p1)
    print(cnt)
    if (cnt %% 10L == 0L) {
      print(paste("save", length(sim)))
      missing <- sim
      save(missing, file=rda)
    }
  }
}
```

Appendix B
Source Code for Study 2

```
# This is a replication of Cai, Yang, & Hansen (2011) simulation study #1.

#options(error = utils::recover)
library(OpenMx)
library(rpf)
library(mvtnorm)

mk.model <- function(model.name, numItems, latent.free) {
  spec <- list()
  spec[1:numItems] <- rpf.grm(factors = 2)

  dims <- (1 + numItems/4)
  design <- matrix(c(rep(1,numItems),
                     kronecker(2:dims,rep(1,4))),
                   byrow=TRUE, ncol=numItems)

  ip.mat <- mxMatrix(name="ItemParam", nrow=3, ncol=numItems,
                     values=c(1.4,1,0),
                     free=c(TRUE,TRUE,TRUE))

  for (ix in 1:numItems) {
    for (px in 1:3) {
      name <- paste(c('p',ix,',',px), collapse='')
      ip.mat@labels[px,ix] <- name
    }
  }
  eip.mat <- mxAlgebra(ItemParam, name="EItemParam")

  m.mat <- mxMatrix(name="mean", nrow=1, ncol=dims,
                    values=0, free=latent.free)
  cov.mat.free <- FALSE
  if (latent.free) {
    cov.mat.free <- diag(dims)==1
  }
  cov.mat <- mxMatrix(name="cov", nrow=dims, ncol=dims, values=diag(dims),
                      free=cov.mat.free)

  m1 <- mxModel(model=model.name, ip.mat, eip.mat, m.mat, cov.mat,
                mxExpectationBA81(
                  ItemSpec=spec,
                  design=design,
                  EItemParam="EItemParam", ItemParam="ItemParam",
                  mean="mean", cov="cov",
                  qpoints=21, qwidth=5),
                mxFitFunctionML())
  m1
}

omxIFAComputePlan <- function(groups) {
  mxComputeIterate(steps=list(
    mxComputeOnce(paste(groups, 'expectation', sep='.'), context='EM'),
    mxComputeNewtonRaphson(free.set=paste(groups, 'ItemParam', sep=".")),
    mxComputeOnce(paste(groups, 'expectation', sep=".")),
    mxComputeOnce(adjustStart=TRUE, 'fitfunction')
  ))
}

g2.mean <- c(1, -.5, 0, .5)
g2.cov <- diag(c(.8, 1.2, 1.5, 1))
correct <- matrix(c(1, 1.4, 1.7, 2, 1.4, 1.7, 2, 1, 1.7,
                    2, 1, 1.4, 2, 1, 1.4, 1.7,
                    .8,1.5, 1.2, 1, 1, .8, 1.5, 1.2, 1.2,
                    1, .8, 1.5, 1.5, 1.2, 1, .8,
                    rep(c(1, .25, -.25, -1), 4)), byrow=TRUE, ncol=16)

groups <- paste("g", 1:2, sep="")

fit1 <- function(seed) {
  result <- list(seed=seed)

  set.seed(seed)

  g1 <- mk.model("g1", 16, FALSE)
  g2 <- mk.model("g2", 12, TRUE)

  data.g1 <- rpf.sample(1000, g1@expectation@ItemSpec,
                        correct, g1@expectation@design)
  data.g2 <- rpf.sample(1000, g2@expectation@ItemSpec,
                        correct[,1:12], g2@expectation@design,
                        mean=g2.mean, cov=g2.cov)

  g1 <- mxModel(g1, mxData(observed=data.g1, type="raw"))
  g2 <- mxModel(g2, mxData(observed=data.g2, type="raw"))

  grpModel <- mxModel(model="groupModel", g1, g2,
      mxFitFunctionMultigroup(paste(groups, "fitfunction", sep=".")),
      omxIFAComputePlan(groups))

  grpModel <- mxRun(grpModel)

  result$cpuTime <- grpModel@output$cpuTime
  result$LL <- grpModel@output$Minus2LogLikelihood
  result$param <- grpModel@submodels$g1@matrices$ItemParam@values
  result$mean <- grpModel@submodels$g2@matrices$mean@values
  result$cov <- grpModel@submodels$g2@matrices$cov@values
  result
}

mc.estimate <- function (bank, sl) {
  example <- bank[[1]][[sl]]
  bias <- matrix(0, nrow=dim(example)[1], ncol=dim(example)[2])
  for (sx in 1:length(bank)) {
    bias <- bias + bank[[sx]][[sl]]
  }
  bias / length(bank)
}

bank <- list()
#setwd("/opt/OpenMx")
rda <- "ifa-cyh2011.rda"
if (file.exists(rda)) load(rda)
for (seed in 1:500) {
  if (length(bank)) {
    if (any(seed == sapply(bank, function (b) b$seed))) next;
  }
  bi <- length(bank) + 1
  bank[[bi]] <- fit1(seed)
  save(bank, file=rda)

  if (0) {
    cur <- bank[[bi]]
    print(cor(c(cur$param, cur$mean, diag(cur$cov)),
                c(correct, g2.mean, diag(g2.cov))))
  }
}

if (1) {
  bias <- c(mc.estimate(bank, 'param') - correct,
            mc.estimate(bank, 'mean') - g2.mean,
            mc.estimate(bank, 'cov') - g2.cov)
  omxCheckTrue(abs(bias) < .061)
}

if (0) {
  require(ggplot2)
  sbank <- bank[300:800]
  df <- rbind(
    data.frame(true=c(correct),
               bias=c(mc.estimate(sbank, 'param') - correct),
               type=c("primary", "specific", "diff")),
    data.frame(true=c(g2.mean),
               bias=c(mc.estimate(sbank, 'mean') - g2.mean), type="mean"),
    data.frame(true=diag(g2.cov),
               bias=diag(mc.estimate(sbank, 'cov') - g2.cov), type="var"))

  df$type <- factor(df$type)
  ggplot(df, aes(true, bias, color=type)) + geom_point(size=3) +
      xlab("true parameter value") +
          ylab(paste("bias (", length(sbank), "replications)"))
}
```