Item Factor Analysis: Everything I told you in the spring was wrong, and current status

Joshua N. Pritikin

Department of Psychology
University of Virginia

17 Oct 2013

Anatomy of an item

Nominal response:

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

Ordinal response:

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

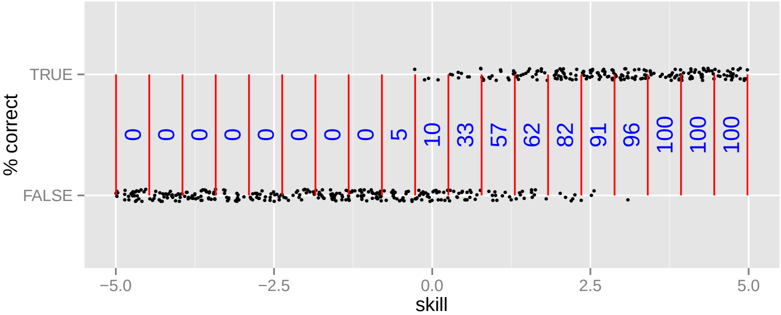
How to analyze?

Treat data as continuous (ratio or interval scale)

Or think more carefully

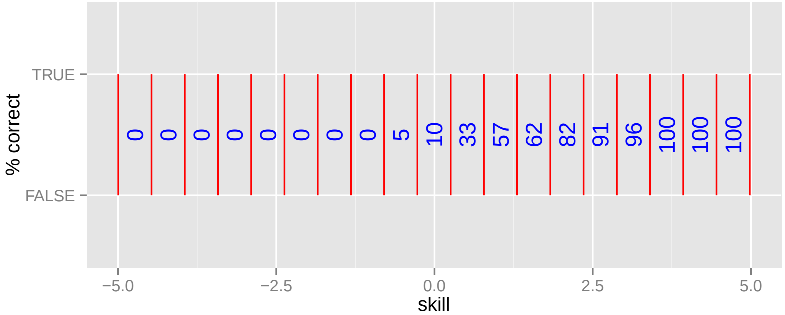
- Assume true skill is known
- Partition responses into bins based on skill

Empirical response plot



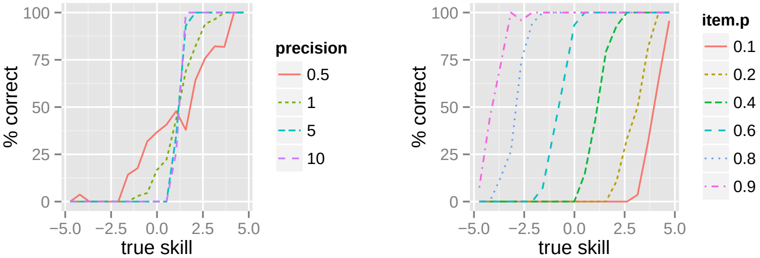
Empirical response plots are constructed by ordering responses by true skill and dividing the data into bins (20 bins, in this example).

Empirical response plot



Ignore individual responses. Voilà! We obtain continuous data conditional on skill.

Sigmoid curve



Plot these conditional response curves for different kinds of items. We usually obtain an S-shape (or sigmoid shape).

See a curve? Parameterize it

What famous function has a similar shape? Why, it looks like the Normal cumulative distribution function (CDF).

The Normal CDF is computationally inconvenient because it includes

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

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).

See a curve? Parameterize it

Additional parameters a and c are introduced in the logistic and become the focus of item characterization,

$$\begin{aligned} \mathrm{P}(\text{pick} = 1|a, c, \theta) &= \frac{1}{1 + \exp(-(a\theta + c))} \\ \mathrm{P}(\text{pick} = 0|a, c, \theta) &= 1 - \mathrm{P}(\text{pick} = 1|a, c, \theta). \end{aligned}$$

In the tradition of item factor analysis (IFA), this is the 2PL item model.

Software

Finding item parameters without knowledge of the true skill requires specialized software.

- ConQuest, \$750
- IRTPRO, \$495
- flexMIRT, $\approx$ \$100 per year

Weaknesses:

- flexibility, customization
- Windows-centric
- non-zero \$ barrier to entry

Open-source Software

ltm (Rizopoulos, 2006) – Has serious bugs. Steer clear.

mirt (Chalmers, 2012) – Promising for traditional IFA analyses

OpenMx (Boker et al., 2011) – New IFA implementation as a module of the OpenMx structural equation modeling (SEM) software.

Rescale the latent distributing during optimization?

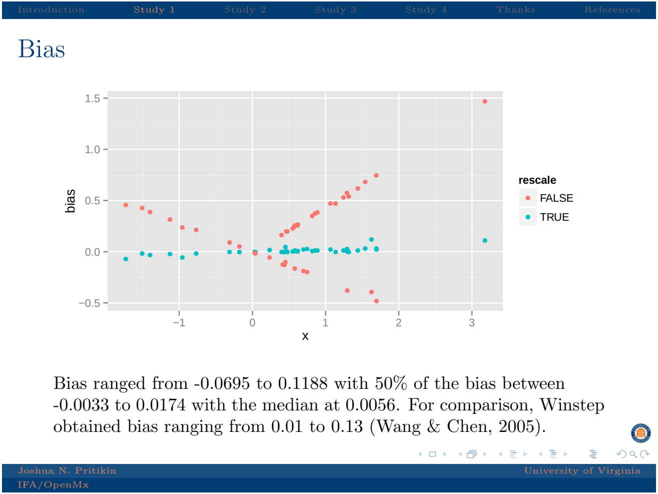
Standardize (rescale) the latent distribution or not?

- 20 2PL items
- 500 persons
- 17 point GH quadrature
- 0% missing
- 500 Monte Carlo replications M
- Rescale or not (Liu, Rubin, & Wu, 1998)

Examine -2LL, $S\!-\!\chi^2$, and bias.

$$\text{bias} = \hat{\theta} - \theta_{true} \text{ where } \hat{\theta} = \frac{1}{M}\sum_{m=1}^{M}\theta_m$$

Rescale the latent distributing during optimization?



Rescale the latent distributing during optimization?

Verdict:

Doesn't replicate with current code.

Rescaling slows down convergence (slightly).

Schilling and Bock (2005) suggested rescaling as a way to speed up convergence of adaptive quadrature.

Does the parameterization matter?

Which model? How many quadrature points?

- 20 2PL items
- 500 persons
- 0% missing
- GH quadrature
- This and subsequent studies are rescaled (Liu et al., 1998)

Traditional parameterization

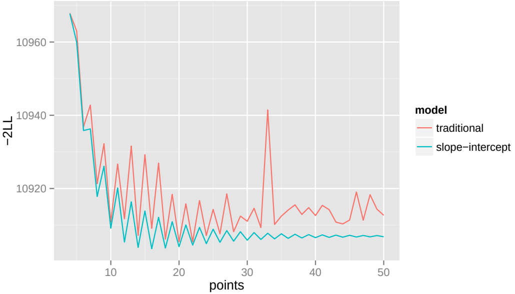
$$\frac{1}{1+\exp(-a(\theta-b))}$$

or slope-intercept form

$$\frac{1}{1+\exp(-(a\theta+c))} \text{ where } b=\frac{c}{-a}$$

Does the parameterization matter?

Likelihood by item model and quadrature points



Does the parameterization matter?

Verdict:

Doesn't replicate with current code.

Prior result due to lack of convergence criteria?

Slope-intercept form does have a smoother likelihood surface that should be easier to optimize.

How does missingness influence bias?

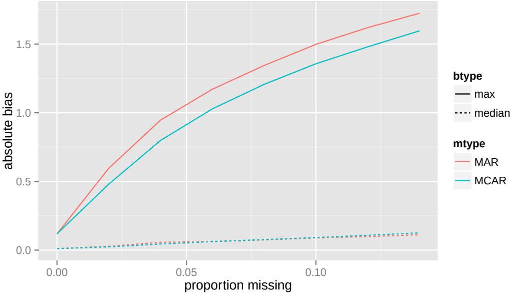
Bias by % missing

- 20 2PL items
- 500 persons
- 17 point GH quadrature
- 500 Monte Carlo replications

For the missing at random condition, data was replaced by NA depending on the first 5 items.

How does missingness influence bias?

Bias by % missing



How does missingness influence bias?

Verdict:

% missing doesn't matter

What is important is the amount of data per parameter.

How does missingness influence bias?

- Newton-Raphson tolerance $= 10^{-7}$
- E-M tolerance $= 10^{-4}$
- Quadrature set to 31 points between -5 and 5
- 20 2PL items
- Two conditions: Missing completely at random (MCAR) and missing at random (MAR)
- MAR operationalized by erasing data in the last 15 items prioritizing by the sum score of the first 5 items.
- Full data consisted of 5000 simulated response patterns
- 500 Monte-Carlo replication

How does missingness influence bias?

Revisiting the Cai (2010) parameter recovery study

Cai (2010b) parameter recovery simulation

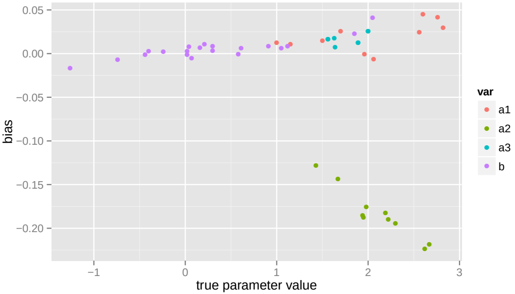
- 20 M2PL items
- 2 primary dimensions
- 4 specific dimensions formed by 4 pairs of item doublets
- $N = 500$ per replication
- 13 point GH quadrature[1]
- 500 Monte Carlo replications

[1] IRTPRO uses a 21 point equally spaced quadrature by default.

Joshua N. Pritikin University of Virginia
IFA/OpenMx

Revisiting the Cai (2010) parameter recovery study

Cai (2010b) parameter recovery simulation



Note: slight pos bias; comparable to Cai (2010b) except for green; *very* slow

Revisiting the Cai (2010) parameter recovery study

Verdict:

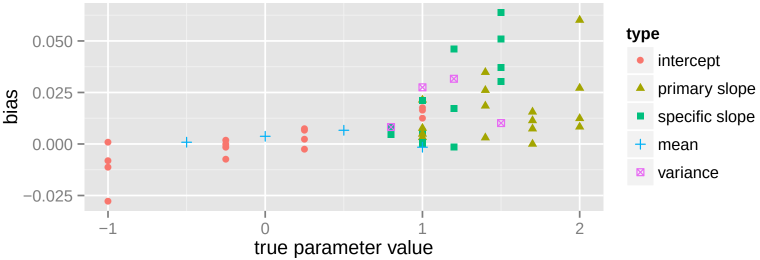
Works now

- OpenMx using a 21 point equally spaced quadrature (identical to Cai, 2010)
- Performance comparable with flexMIRT. On my laptop, 1 randomly chosen replication limited to 1 CPU:

flexMIRT	13s
OpenMx	9.5s

- But flexMIRT has worse starting values (worth 2-3 E-M cycles) and spends an extra 25 E-M cycles to improve the -2LL by 0.0017 beyond what OpenMx can achieve.
- Premature to declare a performance champion

Revisiting the Cai (2010) parameter recovery study



Maximum absolute bias was 0.0603 after 500 replications (indistinguishable from Cai, 2010).

High-priority wish list (from spring)

~~Merge into OpenMx~~

~~Nominal model w/ analytic Newton-Raphson~~

Item parameter standard errors (Cai, 2008)

~~Multiple groups~~

Structural latent trait model

Hierarchical factor model

flexMIRT 1.88 Example 2-4 “Graded Model Combined Calibration and Scoring Syntax with Recoding”

```
library(OpenMx)
library(rpf)
data <- read.table("/opt/OpenMx/models/nightly/data/NCSsim.dat")
head(data)
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17
## 1  5  5  5  5  5  4  5  5  5   5   5   5   4   5   5   4   5
## 2  1  2  2  2  2  3  4  2  1   2   2   1   4   5   2   3   3
## 3  4  4  4  3  4  3  1  1  1   4   4   5   2   4   4   4   4
## 4  4  4  4  4  5  3  5  4  5   4   4   5   3   5   4   3   3
## 5  4  4  4  2  4  3  4  4  1   4   5   4   1   5   4   3   2
## 6  3  2  4  4  4  1  1  4  2   4   4   4   4   3   5   2   3
```

flexMIRT 1.88 Example 2-4

```
spec <- list()
spec[1:18] <- rpf.grm(outcomes = 5)
numItems <- length(spec)
for (c in 1:numItems) {
    data[[c]] <- mxFactor(data[[c]], levels = 1:spec[[c]]@outcom
}
maxParam <- max(sapply(spec, rpf.numParam))
ip.mat <- mxMatrix(name = "ItemParam", nrow = maxParam, ncol = n
    seq(1, -1, length.out = 4)), free = TRUE)
```

flexMIRT 1.88 Example 2-4

```
rpf.rparam(spec[[1]])

##       a      b1      b2      b3      b4
##  1.3265  0.6211  0.1426 -0.1064 -0.4045

ip.mat@values[, 1:2]

##          [,1]    [,2]
## [1,]  1.0000  1.0000
## [2,]  1.0000  1.0000
## [3,]  0.3333  0.3333
## [4,] -0.3333 -0.3333
## [5,] -1.0000 -1.0000
```

flexMIRT 1.88 Example 2-4

```
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)
plan.1loop <- list(
    mxComputeOnce('expectation', context='EM'),
    mxComputeNewtonRaphson(free.set='ItemParam'),
    mxComputeOnce('expectation'),
    mxComputeOnce('fitfunction', maxAbsChange=TRUE,
                  free.set=c('mean','cov')))
inner.plan <- mxComputeIterate(steps=plan.1loop)
plan <- mxComputeSequence(steps=list(
    inner.plan,
    mxComputeOnce('fitfunction', fit=TRUE,
                  free.set=c('mean','cov'))))
```

flexMIRT 1.88 Example 2-4

```
m2 <- mxModel(model="m2", m.mat, cov.mat, ip.mat,
              mxData(observed=data, type="raw"),
              mxExpectationBA81(mean="mean", cov="cov",
                                ItemSpec=spec,
                                ItemParam="ItemParam"),
              mxFitFunctionML(),
              plan)
m2 <- mxRun(m2, silent=TRUE)
m2@fitfunction@result[1,1]


## [1] 140199
```

flexMIRT 1.88 Example 2-4

From flexMIRT:

```
Statistics based on the loglikelihood of the fitted model:
                         -2loglikelihood:   140199.13
      Akaike Information Criterion (AIC):   140379.13
    Bayesian Information Criterion (BIC):   140919.70
```

```
options(digits = 10)
m2@fitfunction@result[1, 1]


## [1] 140199.1317
```

Acknowledgment

- Timo
- OpenMx development team
- Karen
- UVa grad students

& colleagues who I forgot to mention

Questions?

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

Cai, L. (2008). SEM of another flavour: Two new applications of the supplemented EM algorithm. *British Journal of Mathematical and Statistical Psychology*, *61*, 309–329.

Cai, L. (2010). A two-tier full-information item factor analysis model with applications. *Psychometrika*, *75*, 581–612. Retrieved from `http://dx.doi.org/10.1007/s11336-010-9178-0` doi: 10.1007/s11336-010-9178-0

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.

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/`

Rizopoulos, D. (2006). ltm: An R package for latent variable modelling and item response theory analyses. *Journal of Statistical Software*, *17*(5), 1–25. Retrieved from http://www.jstatsoft.org/v17/i05/

Schilling, S., & Bock, R. D. (2005). High-dimensional maximum marginal likelihood item factor analysis by adaptive quadrature. *Psychometrika*, *70*(3), 533–555.