Multidimensional Item Response Theory Workshop in R

Phil Chalmers

York University

February 10, 2015

Overview of Workshop 1

Workshop 1

Primarily about estimating item parameters and diagnosing tests. Workshop 2 will be more about test scoring and explaining variability using IRT.

- Overview of basic and advanced IRT concepts useful for applied item analyses
- Statistical estimation techniques for obtaining optimal parameter estimates in unidimensional and multidimensional models
- Hands-on applied analysis with **mirt**, including model fit and item diagnostic utilities, fitting common and customize item response models, nonparametric estimation of latent trait densities, utilizing different estimation algorithms, computation of standard errors, etc
- Utilizing multiple-group estimation techniques to detect differential item and test functioning

Item response theory

Item response theory (IRT) is a set of latent variable techniques specifically designed to model the interaction between a participants ability, or latent trait, with item level stimuli (difficulty, guessing, etc.)

Three main reasons to use IRT:

- **Model a test** (parameter estimation, diagnostics, dimensionality checking, etc.) in which focus is on the item/population parameters,
- **Explain variability** either in the item properties or persons who were given the test, and
- **Score a test** to obtain estimates of the latent trait(s) for individual participants

Understanding test data

When analyzing test data, we have responses to questions as our primary source of information. Generally, this can be coded numerically:

```
##      Item_1 Item_2 Item_3 Item_4 Item_5 Item_6
## [1,]      0      0      0      3      1      2
## [2,]      1      0      0      3      1      2
## [3,]      0      1      1      3      1      3
## [4,]      0      0      0      2      1      2
## [5,]      0      1      1      3      1      3
## [6,]      1      1      1      3      1      2
```

But what we really want to obtain is some kind of 'scoring' procedure to help us state properties like

- *person 1* $>$ *person 2* $<<$ *person 5* $>$ *person 4*, w.r.t. their ability
- had *person 3* been given a different item we would expect them to have a 90% chance of answering correctly
- Some population of individuals are more likely to answer questions correctly, regardless of their ability (e.g., native versus non-native speaking populations)

What is Item Response Theory?

- Item response theory (IRT) is a set of latent variable techniques specifically designed to model the interaction between a subject's *ability* (i.e., latent trait) and item-level stimuli (difficulty, guessing, etc.)
- Focus is on the **pattern of responses** rather than on composite variables and linear regression theory (i.e., classical test theory), and emphasizes how responses can be thought of in probabilistic terms
- Larger emphases on the error of measurement for **each test item** with respect to particular *ability* levels rather than a global index of reliability/measurement error (e.g., Cronbach's α, McDonald's ω, etc.)
- Widely used in educational and psychological research to study latent variable constructs other than ability (e.g., personality, motivation, psychopathology)

What is Item Response Theory?

Unidimensional or Multidimensional

Most common IRT models are unidimensional, meaning that they model each item with only one latent trait, although multidimensional IRT models are becoming more popular due to their added flexibility.

Unidimensional IRT

Item Response Theory Models

Unidimensional IRT models (dichotomous)

IRT models were originally developed to model how a subject's ability (θ) was related to answering a test item (0 = incorrect, 1 = correct) given item-level proprieties, and how this could be understood probabilistically.

$$P(y = 1|\theta, a, d) = \frac{1}{1 + \exp\left(-(a\theta + d)\right)}$$

- This is the two-parameter logistic model (2PL)[1].
- Given some ability, θ, the probability of positive endorsement is non-linearly related to the item easiness (d) and it's slope/discrimination (a). In canonical form: $\log(P/(1-P)) = a\theta + d$
- The dichotomous Rasch model is realized when the slope parameters are fixed to a constant (usually 1)

[1] Those who are more familiar with the traditional IRT metric, where $a\theta + d = a(\theta - b)$, the a parameters will be the identical for these parameterizations, while $b = -d/a$

IRT trace line

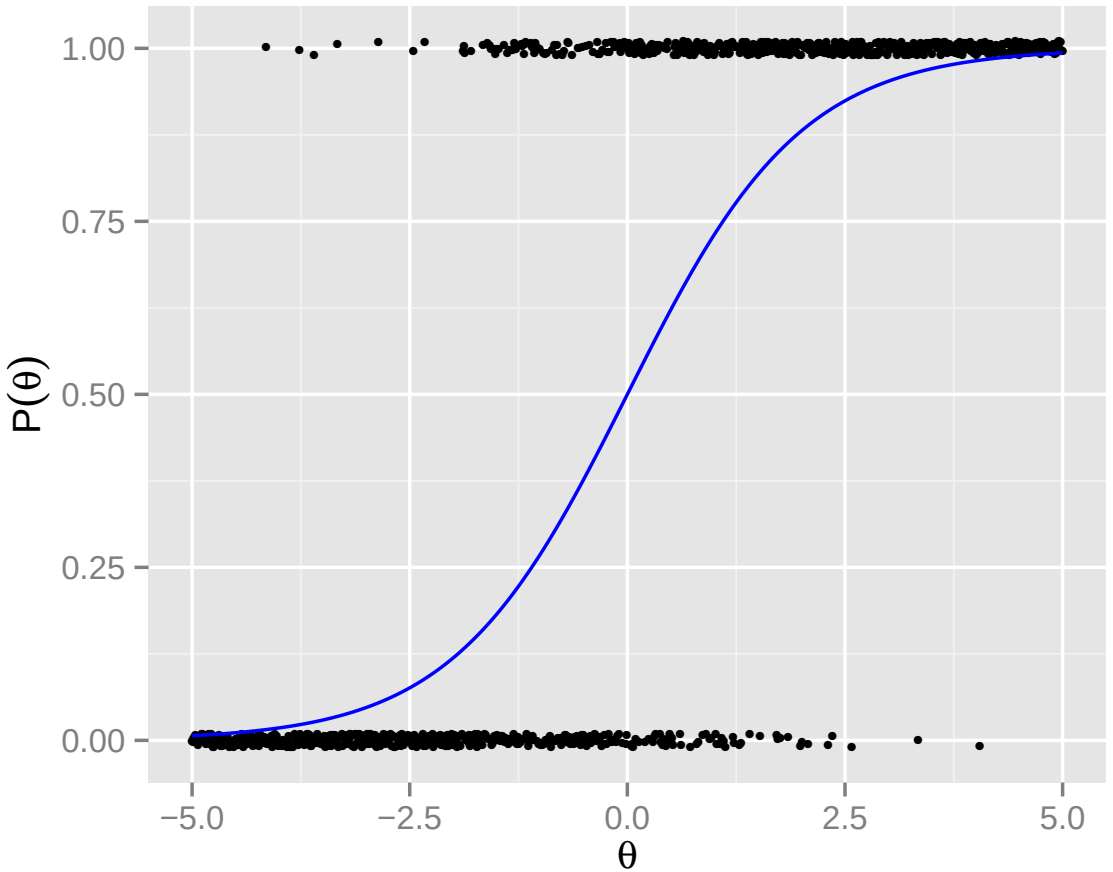


Figure 1: The 2PL model is similar to a logistic regression model; however, in IRT θ is not observed directly.

Unidimensional plots (2PL)

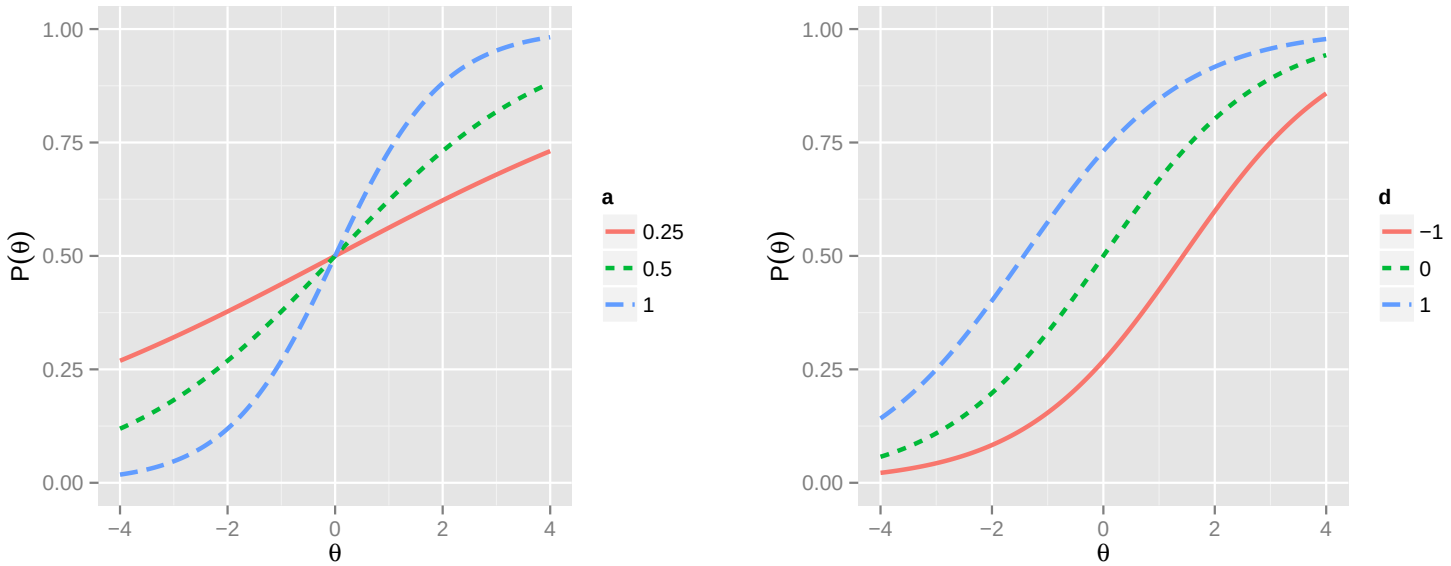


Figure 2: Item response curves when varying the slope and intercept parameters in the 2PL model

Unidimensional IRT models (dichotomous, cont.)

Generalization of the 2PL model are also possible to accommodate for other common testing phenomenon, such as guessing or careless responding effects.

$$P(y = 1|\theta, a, d, g, u) = g + \frac{(u - g)}{1 + \exp\left(-(a\theta + d)\right)}$$

This is the (maybe not so popular, but still pretty cool) four parameter logistic model (4PL), which when specific constraints are applied reduces to the 3PL, 2PL, and Rasch model.

- Given θ the probability of positive endorsement is related to the item easiness (d), discrimination (a), probability of randomly guessing (g), and probability of randomly answering incorrectly (u)
- For psychological questionnaires the lower and upper bounds often have no real rational, and are taken to be 0 and 1, respectively (in clinical instruments they may be justified)

Unidimensional plots (4PL)

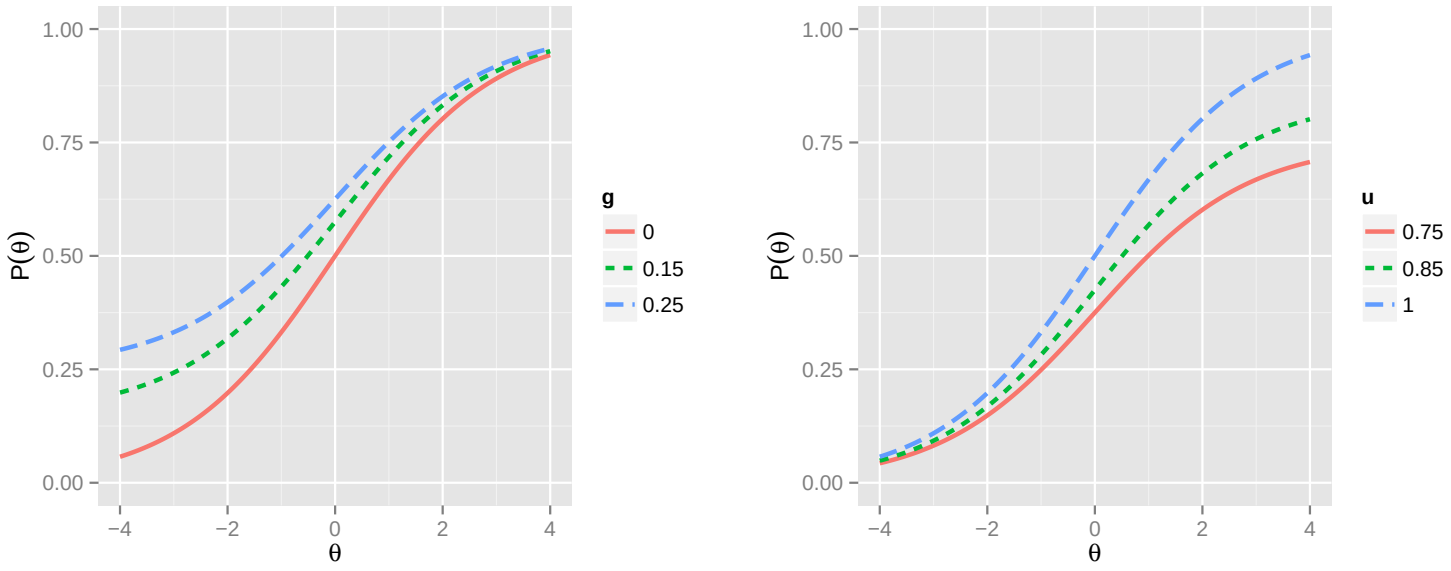


Figure 3: Item response curves when varying the lower and upper bound parameters in the 4PL model

Ideal point models (dichotomous)

Ideal point models are a special type of IRT model that are intimately related to the class of 'unfolding' models in the psychometric literature. They are useful when determining where a person is most likely to be situated in latent variable space when any deviation from their location causes a *decrease* in the response probability.

These types of questions often arise in non-ability based measures (e.g., personality traits, preference ratings, etc).

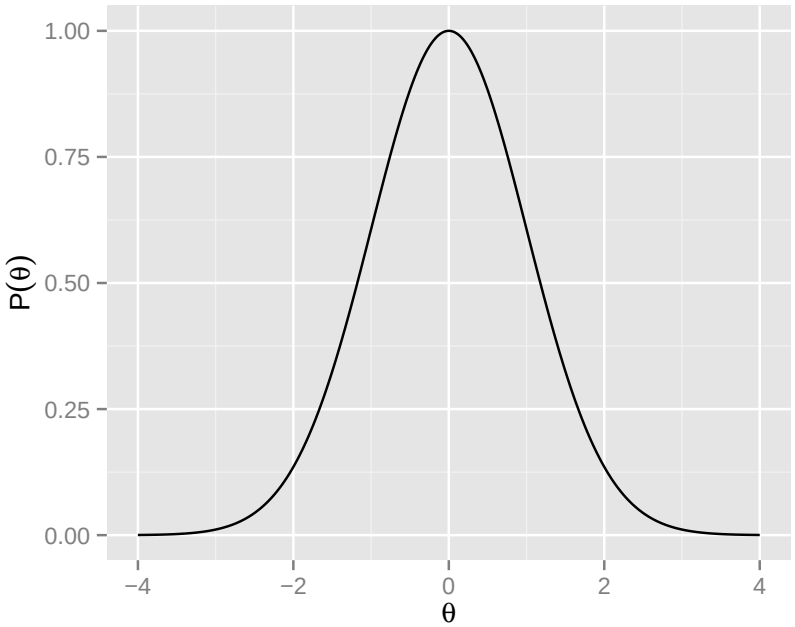
- For example, we ask a participant to agree or disagree with this statement: **I know item response theory fairly well**.
  - If they *agree* to this statement, then there is evidence that they believe they know the material fairly well.
  - However, if they *disagree* with the statement it could be for two distinct reasons: they *do not* know IRT well, **OR**, they know IRT *very well*

Ideal point models

The ideal point models has the form

$$P(y = 1|\theta, a, d) = \exp(-0.5(a\theta + d)^2).$$

- The ideal point model is easily generalized to multidimensional space by including more latent traits (Maydeu-Olivares et al., 2006).



IRT figures with **mirt**

To help understand how the parameterizations in IRT models affect the shape of the probability response curves, I have included an interactive graphical interface to allow the parameters to be modified in real time.

- The interface is shipped with the package by default, and can be called using the following:

```
library('mirt')
itemplot(shiny = TRUE)
```

Unidimensional IRT models (polytomous)

Several different types of polytomous item response models exist for ordinal categories, rating scales, partial credit scoring, unordered categories, and so on.

- Likert scales, for example, are often modeled by ordinal or rating scale/partial credit models. The ordinal/graded response model can be expressed as:

$$P(y_k = k|\theta, \phi) = P(y \geq k) - P(y \geq k + 1),$$

which is simply the difference between adjacent 2PL models (dichotomizing the item at each category, and estimating separate 2PL models).

Unidimensional IRT models (polytomous)

A handful of models are from the so-called 'divide by total' family of IRT parameterizations, such as the (generalized) partial credit model and nominal response model.

- For the generalized partial credit model the ak_k values below are treated as fixed and ordered from 0 to $(k-1)$. This indicates that each successive category is *scored* equally (the ak_k values are often interpreted as scoring coefficients)

$$P(y = k|\theta, \psi) = \frac{exp(ak_k(a\theta) + d_k)}{\sum_{j=1}^{k} exp(ak_k(a\theta) + d_k)}.$$

- ak_k values indicate the ordering of the categories. In nominal models, some ak_k values are estimated to indicate the ordering of the categories empirically.

Unidimensional plots (polytomous)

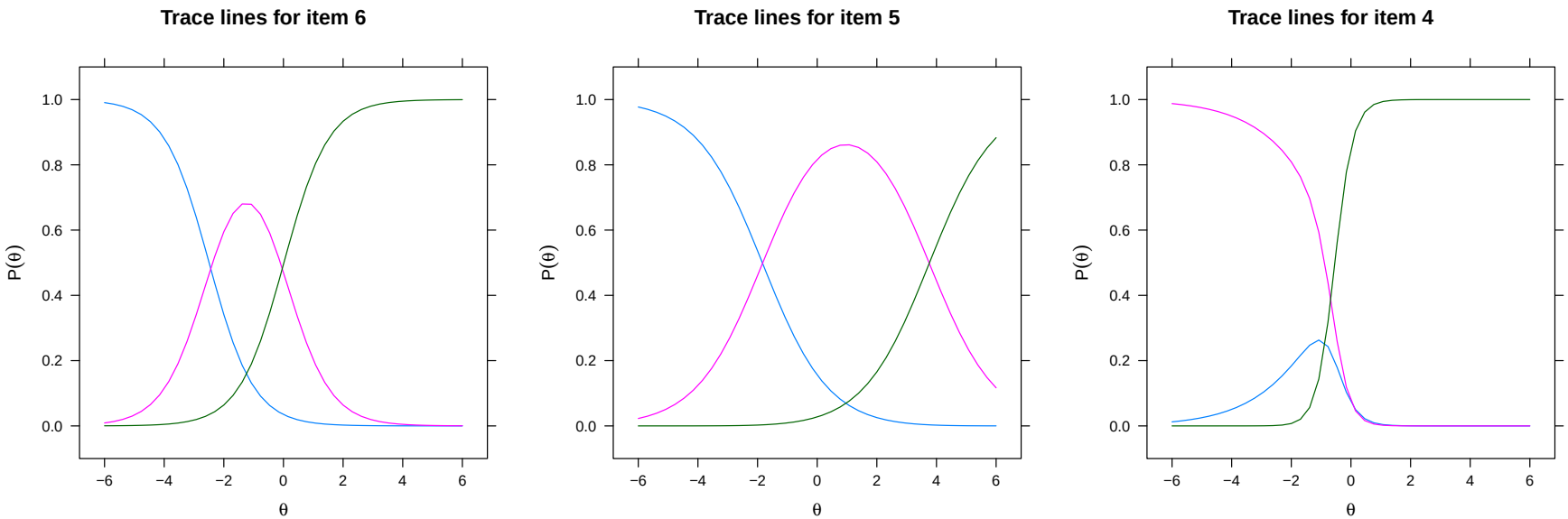


Figure 4: Probability curves for ordinal/graded (left), generalized partial credit (center), and nominal (right) response models

Hybrid of dichtomous and polytomous models

Some response options are worse than others, and sometimes *which* response is selected can be informative. E.g.,

$5 + 6 = ?$

1. 10
2. 11
3. 12
4. 56

Clearly, those who pick 56 *really* do not understand addition. There are multiple approaches to modeling this response phenomemon, and usually this can be detected with the nominal response model.

Hybrid of dichtomous and polytomous models

- Suh and Bolt (2010) introduced a hybrid IRT model for jointly modeling items that have a dichotomous scoring key, but contain additional distractor options (e.g., MC items)
- Contain more information about individuals in the lower θ distribution
- Essentially the model fits a 2-4PL model for the correct response category, and then fits a nominal response model on the remaining 'distractor' options

Item and test scoring functions

It is useful to know what the expected *score* would be given the underlying abilities for both items and tests. Scores are collected by weighting the probability trace lines by their respective category locations. For items,

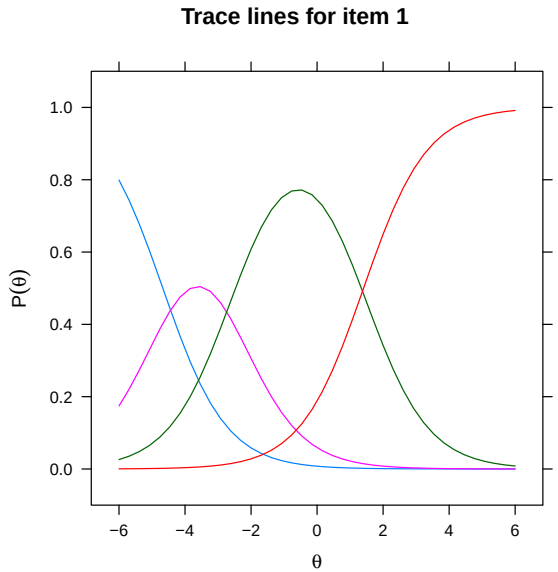
$$S(\theta, \psi) = \sum_{k=0}^{K-1} k \cdot P(y = k|\theta, \psi)$$

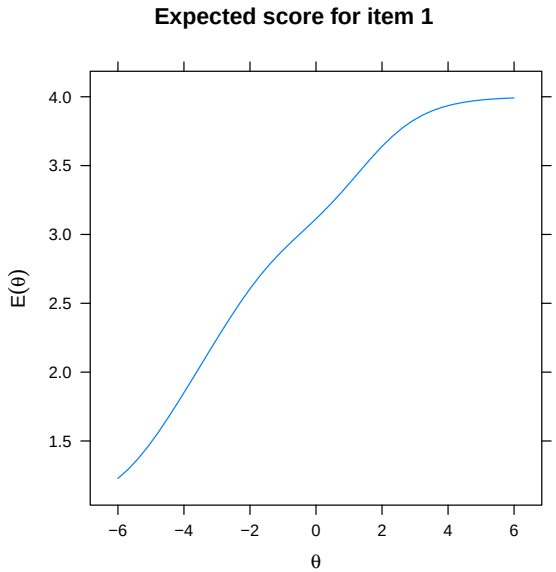
while for the total test,

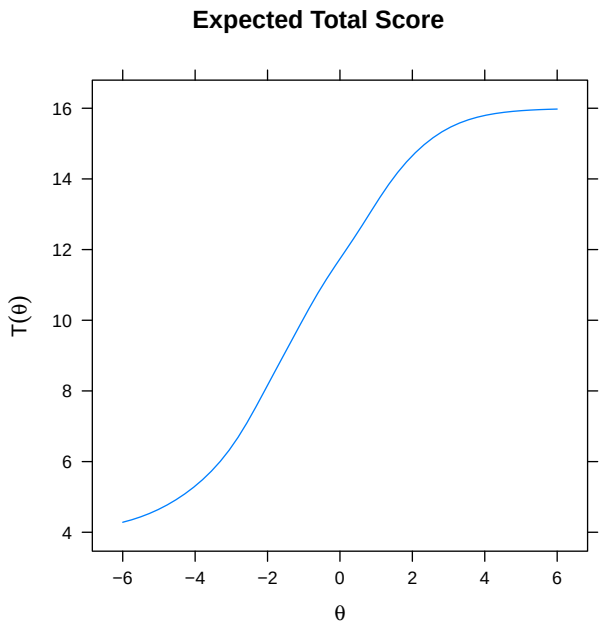
$$T(\theta, \boldsymbol{\psi}) = \sum_{j=1}^{J} S(\theta, \psi)$$

$T(\theta, \boldsymbol{\psi})$ indicates what the expected total score would be given θ.

Item and test scoring functions







First look at **mirt**

First Look at the **mirt** Package

mirt package

Why the **mirt** package?

- Open-source competitor to proprietary software, useful for real data analysis and research (handles thousands of items and millions of participant response), and provides a didactic tool for teaching IRT in classroom settings
- Many flexible parameter estimation features:
  - Mixed (dichotomous and polytomous) IRT model estimation with flexible specifications for unidimensional and multidimensional patterns
  - Linear and nonlinear parameter constraints and bounds
  - Prior parameter distributions
  - Item- and group-level covariates
  - User defined item types for on the fly estimation
  - Nonlinear latent traits and interaction terms

mirt package

- Multidimensional estimation techniques that do not rely on quadrature or joint-ML based schemes (i.e., Metropolis-Hastings Robbins-Monro, quasi-Monte Carlo EM)
- Dimensional reduction capabilities to efficiently estimate two-tier/bi-factor structures. These can help dramatically reduce estimation times while increase precision
- Multiple-group and mixed effects IRT modeling for inspecting group equivalences, modeling item and person-level covariates, and treating different item parameter estimates as fixed or random

mirt package

- Explicit differential item and test functioning support via a likelihood framework (does not require ad-hoc test 'linking' procedures because it can be built into the estimation directly)
- Wide array of plotting features, latent trait estimation, item, person, and model fits statistics, standard error/information matrix calculations
- Customizable prior parameter distributions and integration grids for item and person parameter estimation
- . . . and more!

mirt package

The **mirt** package contains five primary estimation functions, all of which support mixed item formats for multidimensional response models, and each with their own special purpose. They are

- `mirt()` - single group estimation using quadrature (EM) and stochastic MML (MH-RM) estimation
- `bfactor()` - single/multiple group bi-factor or two-tier estimation by using a dimensional reduction EM algorithm, useful when there are multiple packets of independent specific factors (e.g., testlets, longitudinal models)
- `mdirt()` - latent class model estimation (fairly experimental, as there are better packages for such procedures)

mirt package

Two of the modeling functions for including conditional effects (i.e., group membership, multilevel models, test effects, etc) are

- `multipleGroup()` - multiple group estimation, containing useful tools for horizontal and vertical scaling, as well as for detecting DIF
- `mixedmirt()` - mixed effects IRT models for including fixed or random effect covariates at the item and person level. MH-RM estimation engine only

`mirt()` may also include conditional regression effects, but we will cover this more next workshop.

mirt() basics

mirt() requires at minimum two inputs: data, and model.

- data matrix/data.frame that must be structured numerically, and where each row represents a unique individual
- model can be a numerical object specifying the number of factors to extract (similar to how factanal() functions for exploratory factor analysis) or a mirt.model() defined object for more complex factor loading patterns

```
#unidimensional model
mod1 <- mirt(data = data, model = 1)
#two dimensional exploratory model
mod2 <- mirt(data = data, model = 2)

#unidimensional model with mirt.model definition
model <- mirt.model('F1 = 1-5')
mod3 <- mirt(data, model)
```

Possible **mirt** item models

The class of IRT model estimated is chosen based upon the `itemtype` argument passed to `mirt()` and friends. From the `help(mirt)` documentation:

itemtype

type of items to be modeled, declared as a vector for each item or a single value which will be repeated globally. The NULL default assumes that the items follow a graded or 2PL structure, however they may be changed to the following: 'Rasch', '2PL', '3PL', '3PLu', '4PL', 'graded', 'grsm', 'gpcm', 'nominal', 'ideal', 'PC2PL', 'PC3PL', '2PLNRM', '3PLNRM', '3PLuNRM', and '4PLNRM', for the Rasch/partial credit, 2 parameter logistic, 3 parameter logistic (lower or upper asymptote upper), 4 parameter logistic, graded response model, rating scale graded response model, generalized partial credit model, nominal model, ideal-point model, 2-3PL partially compensatory model, and 2-4 parameter nested logistic models, respectively. User defined item classes can also be defined using the createItem function

Additionally, each itemtype model has an associated mathematical definition in the `?mirt` help file.

Generic functions

mirt is designed with object oriented programming in mind, with useful R generic functions that act on estimated model objects.

- `print()` – print the estimated model along with global fit statistics, e.g., G^2, AIC, BIC, etc
- `coef()` and `summary()` – extract unstandardized and standardized (i.e., factor loadings) coefficients, respectively, and optionally rotate the parameters for exploratory models
- `anova()` – comparison between nested models via χ^2, AIC, AICc, BIC, etc
- `plot()` – two- and three-dimensional probability, information, and scoring plots relating to the test as a whole

print()

```r
dat <- expand.table(LSAT7)
lsat_mod <- mirt(dat, 1)
print(lsat_mod)
```

```
##
## Call:
## mirt(data = lsat, model = 1)
##
## Full-information item factor analysis with 1 factor(s).
## Converged within 1e-04 tolerance after 28 EM iterations.
## mirt version: 1.8.2
## M-step optimizer: BFGS
## EM acceleration: Ramsay
## Number of rectangular quadrature: 41
##
## Log-likelihood = -2658.805
## AIC = 5337.61; AICc = 5337.833
## BIC = 5386.688; SABIC = 5354.927
## G2 (21) = 31.7, p = 0.0628
## RMSEA = 0.023, CFI = 0.939, TLI = 0.924
```

coef()

```
coef(lsat_mod, simplify = TRUE)
```

```
## $items
##           a1     d g u
## Item.1 0.988 1.856 0 1
## Item.2 1.081 0.808 0 1
## Item.3 1.706 1.804 0 1
## Item.4 0.765 0.486 0 1
## Item.5 0.736 1.855 0 1
##
## $groupPars
## $groupPars$means
## MEAN_1
##      0
##
## $groupPars$cov
##    F1
## F1  1
```

summary()

```
summary(lsat_mod)
```

```
##            F1    h2
## Item.1 0.502 0.252
## Item.2 0.536 0.287
## Item.3 0.708 0.501
## Item.4 0.410 0.168
## Item.5 0.397 0.157
##
## SS loadings:  1.366
## Proportion Var:  0.273
##
## Factor correlations:
##
##    F1
## F1  1
```

MIRT

Multidimensional Item Response Theory

Multidimensional IRT models

Multidimensional IRT (MIRT) models replace the single θ and a values with vectors $\boldsymbol{\theta}$ and $\mathbf{a}$, respectively. This is analogous to the transition from zero-order logistic regression to multiple logistic regression.

$$P(y = 1|\boldsymbol{\theta}, \mathbf{a}, d, g, u) = g + \frac{(u - g)}{1 + \exp\left[-(\mathbf{a}'\boldsymbol{\theta} + d)\right]}.$$

This model has a very intimate relationship to non-linear factor analysis when $g = 0$ and $u = 1$ (since $logit(P) \approx \mathbf{a}'\boldsymbol{\theta} + d$), and is often called a 'compensatory' model due to the relationship between latent trait scores.

- Similar relationship exists for the graded response model

Multidimensional IRT models

The MIRT extension for the nominal/generalize partial credit model can also readily be understood using the previously declared parameterization.

$$P(y = k|\boldsymbol{\theta}, \psi) = \frac{exp(ak_k(\mathbf{a}'\boldsymbol{\theta}) + d_k)}{\sum_{j=1}^{k} exp(ak_k(\mathbf{a}'\boldsymbol{\theta}) + d_k)}.$$

Again, various ak_k's may be freed to estimate the empirical ordering of the categories (nominal) or treated as fixed values to specify the particular scoring function (gpcm/rating scale).

Multidimensional plots

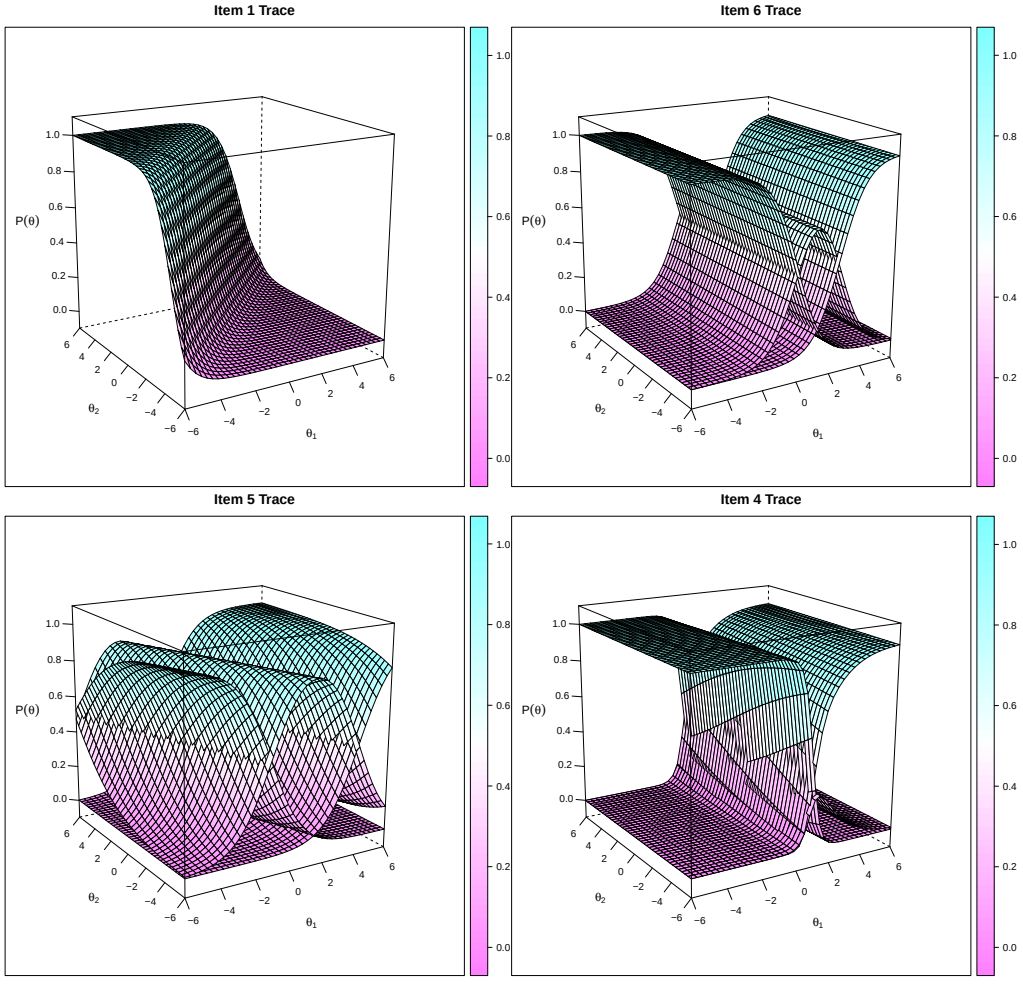


Figure 5: Probability curves for multidimensional 2PL and ordinal (top), generalized partial credit and nominal models (bottom)

Compensatory models

The multidimensional models in the previous graphs are know as 'compensatory' models. The reason for this is that

- A low θ_k parameter on one dimensions does not necessarily entail a low probability of positive endorsement
- High values on adjacent $\theta_{m \neq k}$ can compensate due to the relationship $z = a_1\theta_1 + a_2\theta_2 + d$
- E.g., if $a_1 = a_2 = 1$ and $d = 0$, a participant with the values $\theta_1^{(1)} = -3$ and $\theta_2^{(1)} = 3$ will have *exactly* the same response probability as an individual with the ability values $\theta_1^{(2)} = \theta_2^{(2)} = 0$

Partially compensatory models

Noncompensatory (or partially compensatory) models, on the other hand, are not as affected by high/low θ since they are constructed by multiplying individual 2PL response curves:

$$P(y = 1|\boldsymbol{\theta}, \psi) = g + (1 - g) \prod_{k=1}^{m} \frac{1}{1 + \exp(-(a_k\theta_k + d_k))}$$

This model appears to be appealing from a theoretical perspective in many ability testing situations where the response probabilities should be entirely dependent on adjacent traits.

- E.g., a question that asks how to solve a mathematical problem, but presents the problem in words, will require the subject to have a sufficient *reading comprehension* before being able to measure their *mathematical ability*.
- Unfortunately parameters can be very unstable without highly optimal data conditions (Chalmers & Flora, 2014).

Partially compensatory models

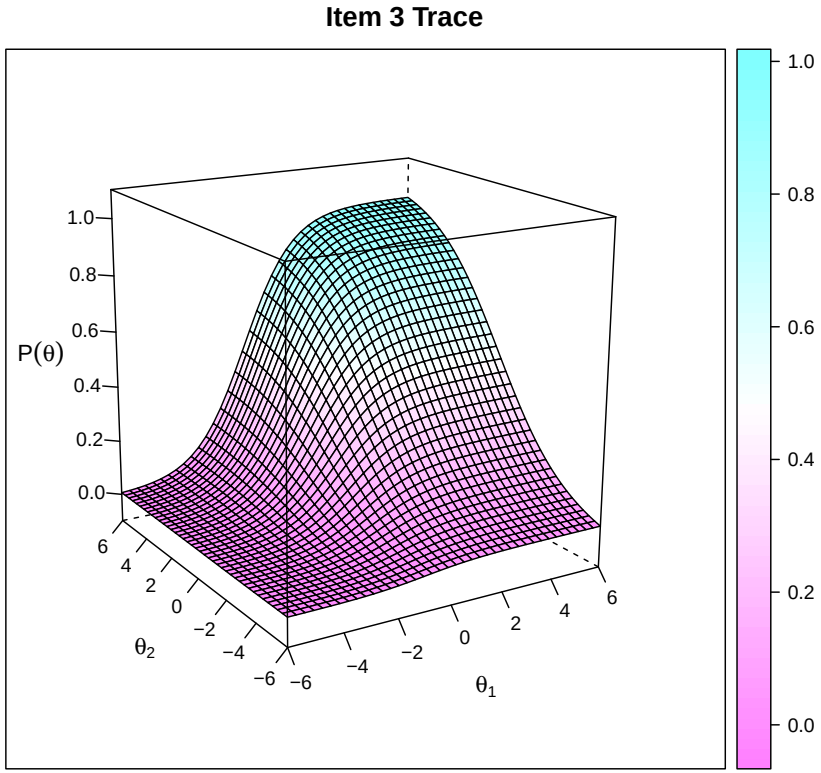


Figure 6: Partially compensatory 2PL model with $a_1 = 1$, $a_2 = 0.5$, $d_1 = 1$ and $d_2 = 0$.

Exploratory and Confirmatory

- As we have seen, MIRT models can often be understood as non-linear extensions of more traditional linear factor analysis methodology, and as such have **exploratory** and **confirmatory** aspects
- For exploratory models, the orientation of the θ axes used to estimate the model are constrained to be orthogonal (no inter-factor correlations), and should be rotated following convergence for better interpretation
- Confirmatory models have no rotational indeterminacy, and are similar to confirmatory FA in structural equation modeling (by definition, unidimensional IRT models are confirmatory)

Confirmatory models are generally specified with the `mirt.model()` function's syntax.

mirt.model() syntax

mirt uses a customized syntax for defining confirmatory patterns (i.e., Q-matrix), and requires calling the `mirt.model()` function. Factor names can be defined by the user, but the keyword `COV` is reserved for specifying which covariance parameters should be estimated.

```
# example of a simple structure with correlated an
# inter-factor correlation between factors 1 and 2, and 1 and 3.
# Factor 1 loads on items 1 to 4 and 6, factor 2 loads
# on 7, 8, 9 and 5, and factor 3 loads on items 10 to 20

model <- mirt.model('
   F1 = 1-4, 6
   F2 = 5, 7-9
   F3 = 10-20
   COV = F1*F2, F1*F3')

cmod <- mirt(dat, model)
```

mirt.model() syntax

The syntax definitions can contain other keyword elements as well that are useful for specifying equality constraints, prior parameter distributions, polynomial trait combinations, etc. Also supports using item names instead of index locations.

- CONSTRAIN and CONSTRAINB – parameter equality constraints within and between groups
- PRIOR – specify prior parameter distributions (e.g., normal, log-normal, beta)
- START – specify explicit start/fixed parameter values

```
model2 <- mirt.model('
    Theta = 1-10
    (Theta * Theta) = 2,4,6,8,10   ## quadratic factor

    ## constrain first factor slopes to be equal
    CONSTRAIN = (1-10, a1)

    ## N(0,1) prior on d for item 2,3,5, N(0,0.5) for item 4
    PRIOR = (2-3, 5, d, norm, 0, 1), (4, d, norm, 0, 0.5)')
```

Diagnostics

Test, Item, and Person Diagnostics

Diagnostics

Unfortunately, statistical models may not agree well with the empirical data. Knowing how well our model and items fit within our tests is a very important topic. Diagnostics help us find (and possible fix) potential problems, and help us judge the usefulness of the model.

Empirical problems can areas for many reasons, and **mirt** offers a few useful tools to help diagnose issues at the

- test-level – providing global fit measures
- item-level – checking how well each item fits within the test, and whether there are residual interdependencies between items
- person-level – same as items, but with respect to participants

Test Diagnostics

Global fit statistics are useful to describe how well the model fits overall, but are less useful at diagnosing specific modeling problems. In IRT, global fit stats are possible with the G^2 statistic, but this becomes impractical very quickly due to data sparseness.

$$G^2 = 2\left(\sum_{l}^{s} r_l log_e\left[\frac{r_l}{N\tilde{P}_l}\right]\right)$$

To circumvent the sparseness issue, **mirt** implements the M2 and M2* family of statistics (Maydeu-Olivares & Joe 2006) that are based on the second order marginals of the item covariances.

- Fit statistics are accessible with the `M2()` function. Different fit statistics are also available when passing `method = 'EAPsum'` to `fscores()`.

Test Diagnostics

- The M2 family of fit statistics are intimately related to the fit stats in structural equation modeling, while the `fscores()` approach is based on reducing the sparse response patterns to residuals based on total scores
- Both provide χ^2 type model fit testing for dichotomous and polytomous items by collapsing the extremely sparse data-tables into more manageable marginals
- Related statistics are also available, such as RMSEA, SRMSR (and residual covariance matrix), CFI, TLI, etc

I generally find `M2()` to be more useful, especially for multidimensional models (where the total scores are less meaningful).

Item Diagnostics

There are generally two classes of item diagnostic tools: detecting residual covariation between items, and judging the overall fit of an item within a test.

Covariance-based residuals are available through the `residuals()` function, and include

- local dependence (Chen & Thissen, 1997) statistics (χ^2 and G^2 variants), and
- Q3 statistic (Yen, 1984)

Generally these are χ^2 variants, but may be standardized for easier interpretation. Mostly used for diagnosing multidimensionality, and can return the complete tables of bivariate residuals for more thorough inspection.

Item and Person Diagnostics

Single item/person fit statistics are available through the `itemfit()` and `personfit()` functions. Several options are available.

- For Rasch specific models, the popular *infit* and *outfit* stats are computed (values close to 1 are considered good, and come with an associated z value)
- Z_h statistics printed for all models (Drasgow, Levine and Williams, 1985); values greater than 0 indicate a better fit than expected, less than zero indicate worse

Item and Person Diagnostics

- X^2 is a χ^2-type statistic based on collapsing across the expected probability space (only reasonable for unidimensional models). Plots may also be drawn, if requested (`itemfit()` only)
- S-X2 is a different χ^2 statistics based on conditioning on the raw sum-score, are also available for uni- and multidimensional models (`itemfit()` only)

Item and person fit statistics are generally considered 'two-step' procedures, in that they require plausible estimates for the θ values.

Item Diagnostics

After items are flagged as not fitting well (or the whole model is flagged), it can be helpful to inspect the items further:

- View the patterns of misfit with the observed versus expected tables (M2, S-X2, local dependence, etc)
- Modify the item types to determine if that helps fix the problem (perhaps should have fit a 3PL instead of 2PL)
- Fix a more flexible IRT model, such as the multidimensional nominal response model, to get a better idea of how the item categories are functioning
- Using non-parametric smoothing techniques for more exploratory approaches (such as from the **KernSmoothIRT** package)

Exercise

Exercise

Now is a good time to check out several of the examples in the `mirt()` function, look through the HTML/pdf documentation, and review the examples demonstrated so far in this talk. After that, you should be able to complete the Exercise found in `Exercise_01.html`

Estimation

Model Estimation

Model estimation

IRT item parameters are estimated by maximizing the observed likelihood

$$L(\mathbf{\Psi}|\mathbf{Y}) = \prod_{i=1}^{N} \left[\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} L_{\ell}(\mathbf{y}|\mathbf{\Psi}, \boldsymbol{\theta}) g(\boldsymbol{\theta}) d\boldsymbol{\theta} \right].$$

- Evaluating the integrals results in the so-called marginal maximum likelihood method (MML) since the θ parameters are integrated out of the equation
- Maximizing this equation directly quickly becomes infeasible as the number of items grow

Model estimation

EM algorithm can be used instead to capitalize on a more manageable complete-data likelihood structure, where each item (largely) can be updated independently.

- Using the likelihood equation, evaluate each response pattern over the grid of $\boldsymbol{\theta}$ values
- Given the marginal evaluated response pattern, collect what the 'expected' table of collected response patterns should look like across the grid of values (e.g., given $\boldsymbol{\theta} = 1$, we might expect to see $r_0 = 199.54$ and $r_1 = 800.66$ for item 1)
- Using the expectation table, 'maximize' the item parameters *as if the expectation table was what was really observed*, using the $\boldsymbol{\theta}$ grid as the predictor variable

Pros and cons of EM

- Observed-data information matrix not available and must be approximated in other ways (e.g., S-EM, Monte Carlo, MH-RM, etc). SE's also not available as a consequence
- Effectively this approach removes the problem of maximizing all the parameters at each iteration with the cumbersome observed-data likelihood (Bock and Aitkin, 1982)
- Missing data posses no problem to the ML estimator, and uses all available data (full-information)

This is the default estimation method in `mirt()` and `multipleGroup()`.

Unfortunately . . .

Every new θ in the model requires a new integral to be evaluated.

- The difficult task now is to evaluate the likelihood equations numerically, which requires high dimensional integration by quadrature or simulation methods
- Standard quadrature techniques become intractable as the dimensions increase since the number of quadratures required increases exponentially
- Quasi-Monte Carlo and adaptive integration methods have been used to circumvent this integration problem, but generally only work well for a moderate number of dimensions (e.g., 3–5)
- Fully Monte Carlo methods exist, however these come at the cost of longer estimation times and often high computational demand, especially if a pure Bayesian framework is adopted

Stochastic MML estimation

An alternative approach is to capitalize on the complete-data likelihood function,

$$L(\boldsymbol{\Psi}|\mathbf{Y}, \boldsymbol{\theta}) = \prod_{i=1}^{N} L_\ell(\mathbf{y}_i|\boldsymbol{\Psi}, \boldsymbol{\theta}_i) g(\boldsymbol{\theta}_i|\mu, \boldsymbol{\Sigma}),$$

by imputing plausible values for the missing random effects.

- Obtain 'known' values for $\boldsymbol{\theta}$ and maximize this function instead
- This approach somewhat familiar to the joint ML framework. In joint ML, estimates of $\boldsymbol{\theta}$ are computed, item parameters updated given new $\boldsymbol{\theta}$, then $\boldsymbol{\theta}$ updated again given new item parameters, *ad nosium* until all the parameters stopped moving by some tolerance
- Joint ML estimation requires some very unconventional controlling mechanisms to facilitate convergence, and likely is only viable for Rasch models (suffers from the Neyman-Scott problem)

MH-RM algorithm

Metropolis-Hastings Robbins-Monro (MH-RM) algorithm works well in this situation since it deals with the random variables appropriately.

- Use an MH sampler to obtain $\hat{\boldsymbol{\theta}}$ values, and treat values as provisionally 'known'
- Update parameters using standard numerical optimization methods (e.g., Newton-Raphson) with 1 iteration
- Repeat a number of times to complete a burn-in period so that the solutions is bouncing around the ML location
- Continue, but use the Robbins-Monro noise cancellation method to help remove the sampling error borne from the MH draws

MH-RM algorithm

- Consequence of the estimation is that the parameter information matrix can be easily approximated
- Original work suggested approximating the information matrix from the estimation history, but I have found that approach leads to extremely bad approximations if the number of iterations was too low, or parameter variability was too high
    - **mirt** computes a separate stage for the information matrix by keeping the estimates fixed at their ML locations
- Unfortunately, the log-likelihood must also be computed by further stochastic means (but can be run in parallel)

The MH-RM scheme extends to other random effects as well (as we will see in Workshop 2).

Special EM models: The bi-factor and two-tier models

Special type of confirmatory model in which there are multiple 'packs' of uncorrelated specific factors is known as the bi-factor or two-tier model

- Have especially simple estimation forms because the EM algorithm can be rearranged to create a *highly* reduced integration problem for the item packets
- All specific factors are integrated with only one quadrature grid

```
# general factor, 2 specific factors loading on items 1-3, and 4-6
model <- c(1,1,1,2,2,2)
bmod <- bfactor(data, model) # 2 dimensional integration

# model is equivalent to the following in mirt(),
# using standard integration (3 dimensions)
model <- mirt.model('
    G = 1-6
    S1 = 1-3
    S2 = 4-6')
mmod <- mirt(data, model)
```

Special EM models: The bi-factor and two-tier models

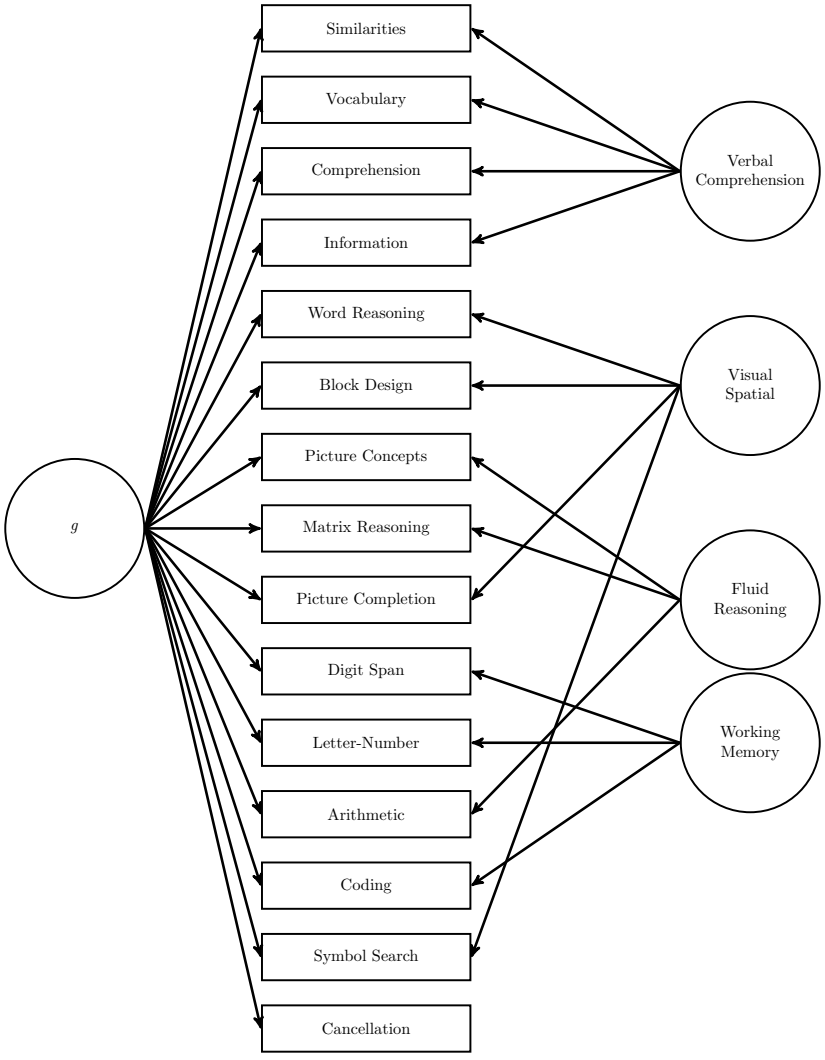


Figure 7: Bi-factor model that requires two dimensions for integration.

Package Specifics

Specific Functions and Utilities in `mirt`

Package specific functions

Functions more specific to the package.

- `fscores()` – compute EAP, EAP for sum scores, MAP, WLE, ML factor scores, plausible values (more on this next workshop)
- `itemplot()` – individual item plots (information, trace lines, confidence envelopes, etc.)
- `wald()` – testing parameter hypotheses with the Wald tests
- `DIF()` and `DTF()` – statistics for testing differential item and test functioning using likelihood-ratio and Wald tests
- `createItem()` – create a user defined item type which can be estimated from the data
- `simdata()` – simulate IRT data from given parameters

Package specific functions

. . . and many more!

- `mirtCluster()` – define a `parallel` object useful for speeding up computations. Useful when computing Monte Carlo log-likelihoods, factor scores, S-EM standard errors, etc.
- `mod2values()` – create `data.frame` object of item parameters
- `averageMI()` – multiple imputation averaging of plausible values using Rubin's method
- `imputeMissing()` – given a converged model, impute plausible response values for the missing data. Useful for obtaining approximate fit statistics not defined when missing data are present
- `key2binary()` – expand collapsed tabulated data table to full dataset
- `extract.group()`, `extract.item()` – internal extraction methods
- `probtrace()`, `iteminfo()`, `testinfo()`, `expected.item()` – lower level item and test statistics from estimated models

simdata()

You might see this function used a lot in the package examples. The function simulates plausible responses given the IRT models and θ values (both of which can be supplied). Requires slopes, intercepts, and the itemtype.

```
# Unidimensional nonlinear factor pattern
theta <- rnorm(2000)
Theta <- cbind(theta, theta^2)

itemtype <- rep('dich', 6)
a <- matrix(c(
.8,.4,
.4,.4,
.7,.4,
.8,NA,
.4,NA,
.7,NA), ncol=2, byrow = TRUE)
d <- matrix(seq(-2.5, 2.5, length.out = 6))

nonlindata <- simdata(a, d, 2000, itemtype=itemtype, Theta=Theta)
```

Parameter standard errors/information matrix

Several methods exist in the package for computing the parameter information matrix for the estimated parameters, which when inverted yields asymptotic covariance matrix. These are passed to the `SE.type` argument:

- `crossprod`, `Louis`, and `sandwich` – Cross-product approximation, exact observed information matrix, and sandwich covariance matrix estimate (`crossprod` is the default because it is very cheap to compute)
- `SEM`: supplemented-EM, computes proportion of missing information from (unaccelerated) EM history. Supports parallel computing
- `BL` and `Fisher`: Bock and Leiberman (1970) approach to obtain observed information matrix and expected information matrix (Fisher). Not recommended when a moderate to large number of parameters are estimated
- `MHRM`: the MH-RM approach to estimating the parameter information matrix

Parameter standard errors only

If parameters are not theoretically symmetric due to estimation bounds (e.g., $0 \geq g \geq 1$) then the quadratic approximations from information matrices may not be appropriate. Instead, you could use

- `boot.mirt()` – for bootstrapped confidence intervals, or
- `PLCI.mirt()` – for profiled-likelihood confidence intervals

Both functions support parallel computing, however `PLCI.mirt()` supports estimation of specific parameters (no need to compute often large and potentially unstable information matrix).

Miscellaneous

Many other things are possible in the estimation engine, including

- `optimizer` – changing the default optimizer. Default is the `BFGS` algorithm, but other are possible to impose bounds (`L-BFGS-B`) or include non-linear parameter constraints (`solnp` or `alabama`)
- `method` – default is `EM` quadrature integration, but may also be quasi-Monte Carlo integration (`QMCEM`) or `MHRM`
- `survey.weights`, `accelerate`, `technical` – for survey weights, changing the EM acceleration scheme, and passing lower-level technical arguments (including modifying latent distribution functions, integration grids, and so on)

Evaluated code

- For convienience, code has been evaluated using `knitr` and hosted online
  - Github wiki (`https://github.com/philchalmers/mirt/wiki`).
- Documentation and examples
- User contributed examples
- Exercises from previous workshops

Multiple Groups

Multiple Group IRT

Multiple Group models

Multiple group analysis (MGA) takes into account empirical grouping clusters that are thought to behave differently to the response data. For instance, items may be more difficult for one group or another, may have unequal slopes, etc., and these play a key role in determining the 'fairness' of a test.

- MGA has two extreme ends: completely ignore group membership (aka, a single group) or completely separate the data according to membership (i.e., multiple single groups)
- MGA becomes useful when models lie somewhere in the middle of these extremes, where we try to find a simpler model than strict independence while being mindful of population differences

Multiple Group likelihood

The log-likelihood equation that is evaluated for these models is

$$LL_{total} = LL_{G1} + LL_{G2} + \cdots + LL_{Gn}$$

Parameters can therefore be constrained to be equal across group, or freely estimated, and allows for nested model comparisons.

- Special cases of MGA result in differential item functioning (DIF), where items function differently depending on the group
- DIF also leads to differential test functioning (DTF), a further extension of the DIF principle, but at the test level

Multiple Group models

In **mirt**, the `multipleGroup()` function is used for MGA and defaults to the completely independent groups approach.

- `invariance` argument has keywords to constraint or relax various parameters, such as '`slopes`', '`intercepts`', '`free_means`', etc.
- `mirt.model()` syntax arguments with the `CONSTRAINB` keyword is also very useful to constrain parameter between the groups to be equal for testing

Nested comparisons

Here's an example where we set the slopes to be across groups (Wald tests with the `wald()` function may be useful here too if the information matrix is computed).

```
mg1 <- multipleGroup(dat, model = 1, group = group, verbose = FALSE)
mg2 <- multipleGroup(dat, model = 1, group = group,
                     invariance = 'slopes', verbose = FALSE)
anova(mg2, mg1)
```

```
##
## Model 1: multipleGroup(data = dat, model = 1, group = group, invaria
##     verbose = FALSE)
## Model 2: multipleGroup(data = dat, model = 1, group = group, verbose
```

```
##        AIC     AICc    SABIC      BIC    logLik   X2  df      p
## 1 29709.14 29709.44 29783.94 29860.20 -14830.57  NaN NaN    NaN
## 2 29711.20 29711.67 29804.70 29900.02 -14825.60 9.94   6 0.1272
```

MG invariance

The `invariance` argument provides a quick way to define equality constraints across all groups simultaneously, and also allows the estimation of group-level hyper parameters (e.g., latent means and variances).

- `free_means` – for freely estimating all latent means (reference group constrained to a vector of 0)
- `free_varcov`, `free_var`, `free_cov` – for freely estimate elements of the variance-covariance matrix across groups (reference group has variances equal to 1 by default)
- `slopes` – to constrain all the slopes to be equal across all groups
- `intercepts` – to constrain all the intercepts to be equal across all groups

Additionally, specifying specific item names (from `colnames(data)`) will constrain all freely estimated parameters in the specified item(s) to be equal across groups.

Differential item functioning

DIF is a widely studied area in IRT to detect potential bias in items across different populations. Formally, when

$$P_{focal}(k = K|\theta) \neq P_{reference}(k = K|\theta)$$

then the item is said to demonstrate DIF.

- DIF tests generally require that groups are 'equated', either by ad-hoc linking methods or by providing a set of anchor items to link the θ metrics during estimation
- Different types of DIF exist, but largely these have been grouped into uniform and non-uniform DIF
    - many different methods have even been coded into R already; see the **difR** package

Differential item functioning

mirt supports two DIF approaches based on maximum-likelihood theory: the Wald test, and the likelihood-ratio test. DIF testing may be run manually through `multipleGroup()` or through the more automated testing function `DIF()`.

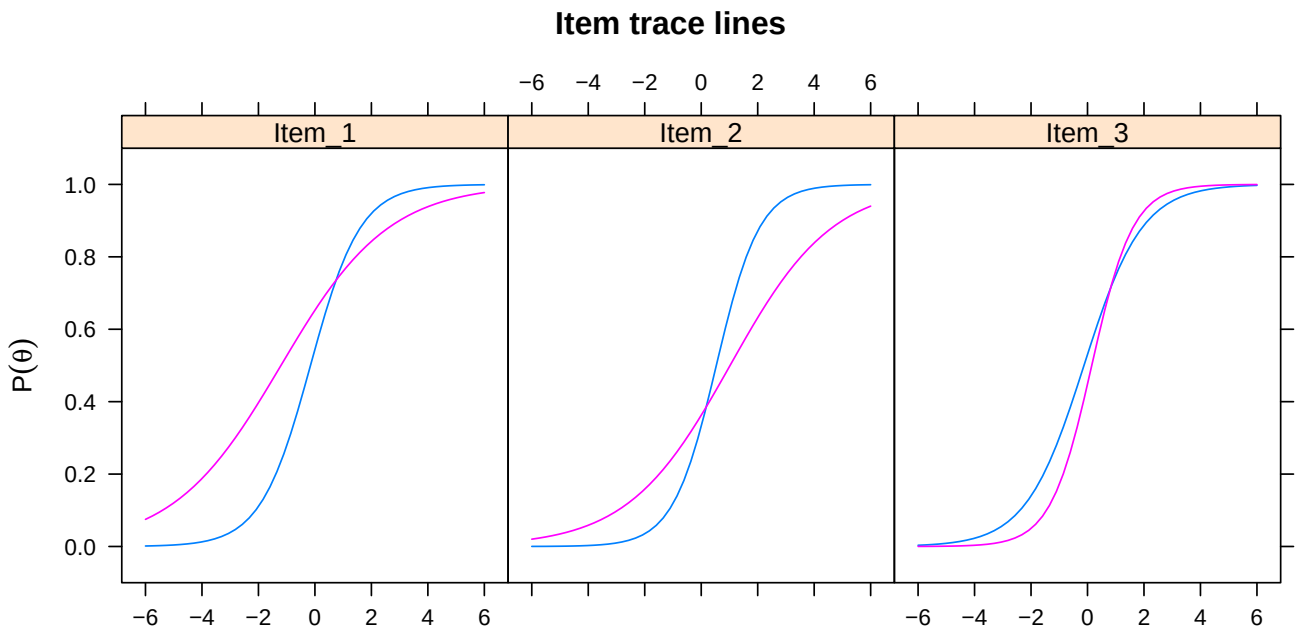
- DIF, from a ML framework, requires that a number of 'anchor' items have been pre-selected (as small number of invariant items), and that the group hyper-parameters are freed for all but one group. This properly 'equates' the groups to remove population differences
- Constrains are added or removed, depending on the starting model, and tested to determine whether the model improves/gets worse
- Items requiring free parameters across groups are said to contain DIF

Differential item functioning

```
#test a1 and d for DIF (2 df)
DIF(constrainedMG, which.par = c('a1', 'd'), scheme = 'drop')

#test d for DIF (1 df) with Wald for items 10 to 15
DIF(constrainedMG, which.par = c('d'), scheme = 'drop',
    Wald = TRUE, items2test = 10:15)

# plot items showing DIF
DIF(constrainedMG, which.par = c('a1', 'd'), plotdif = TRUE)
```



Differential test functioning

DIF is great for detecting biased measurements between groups, but what if we are interested in how DIF affect the test as a whole?

- DIF effects may be very small and make little to no difference when scoring the test
- Some DIF effects may be in opposite directions, and therefore may actually cancel out at the test level (e.g., Item 1: $d_{G1} = 1$, $d_{G2} = 0$; Item 2: $d_{G1} = 0$, $d_{G2} = 1$)
- With more complicated types of DIF this is harder to witness directly, and therefore test statistics and plots are required

Differential test functioning

- Previously proposed DTF statistics were less than satisfactory (e.g., Raju et al., 1995). So, I came up with some new ones (Chalmers, Counsell, Flora, in press).
- Based on the test scoring functions, which in turn are built from item scoring functions
- Statistical variability collected from variability in the parameter estimates through a multiple imputation technique to account for the nonlinear/non-smooth functional form

Differential test functioning

Two statistics proposed: a signed and unsigned DTF stat (sDTF and uDTF), to account for scoring cancellation and overall area differences between test scoring curves

$$sDTF = \int \left(T_{reference}(\theta, \boldsymbol{\psi}) - T_{focal}(\theta, \boldsymbol{\psi})\right) g(\theta)$$

$$uDTF = \int |T_{reference}(\theta, \boldsymbol{\psi}) - T_{focal}(\theta, \boldsymbol{\psi})| g(\theta)$$

where $\int g(\theta) = 1$ and $g(\theta) = C$.

- sDTF can be evaluated at single locations along θ as a diagnostic tool

Differential test functioning

Empirical example showing DTF ($sDTF = 0.629$, $p < .002$).

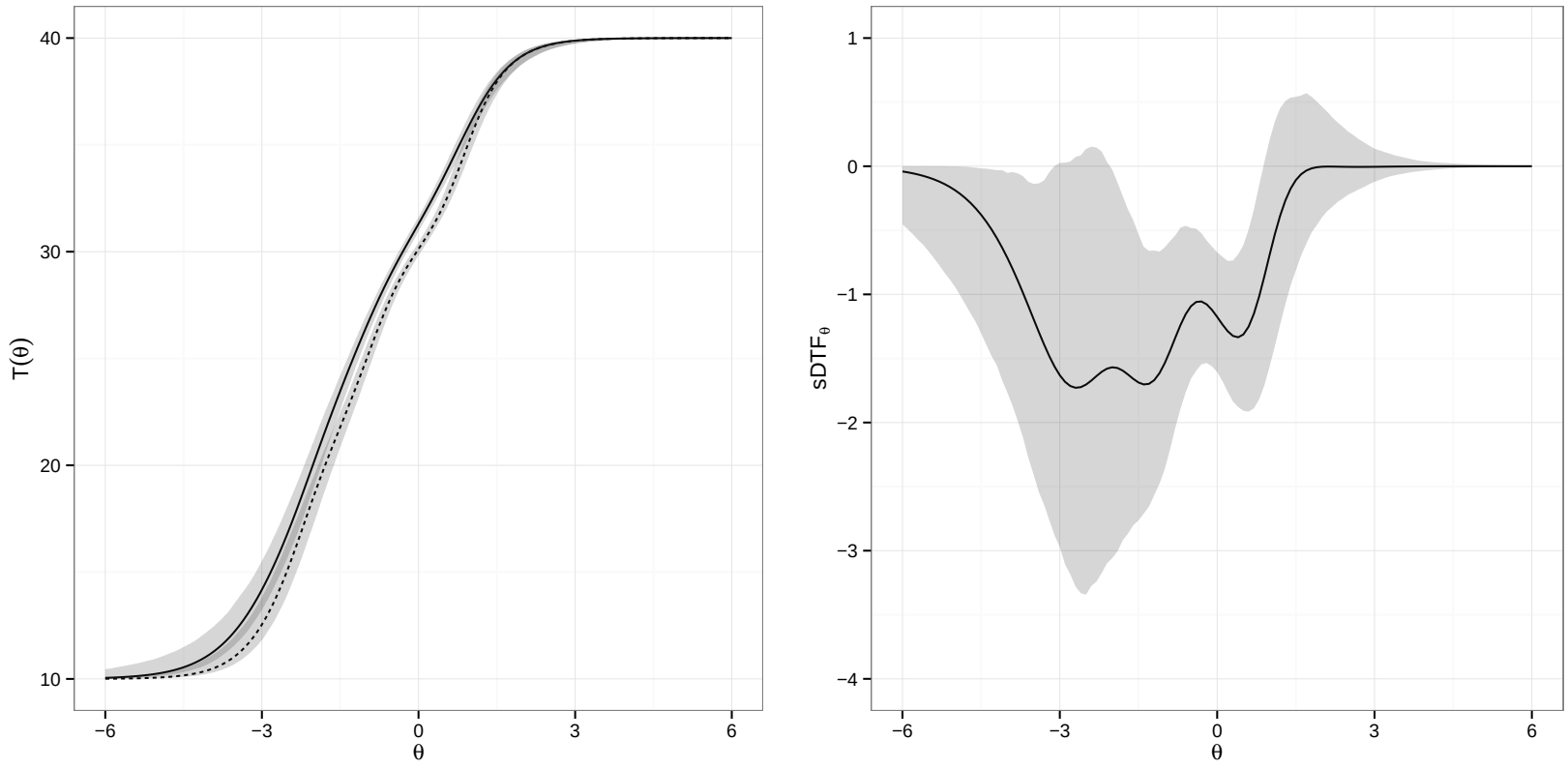


Figure 8: Total score plot with 95% confidence intervals around reference and focal group (left), and variability of sDTF statistic across range of θ (right).

DTF simulations

Two simulations set up: 3PL and graded response models with various design effects:

- Varying sample size (500, 1000, 3000), DIF size (0.0, 0.5 and 1.0), test size (30, 40, and 50), parameters containing DIF (slopes, intercepts, slopes and intercepts), and number of items containing DIF (4, 8, and 12 in the 3PLM design, and 4, 6, and 8 in the GRM design)
- Result? Nominal Type I error rates when no DTF was present in multiple IRT models, and increasing power under many conditions of DTF (uDTF better and differences in slopes, sDTF better at difference in intercepts)

Exercise

Exercise

Exercises pertaining to multiple group estimation and additional mirt functions are available in `Exercise_02.html`.

End of Workshop 1

This is the end of Workshop 1 (yay!). Just to review what we learned:

- Basic properties of IRT (trace-lines, models, expected score functions, etc)
- Item-types that are commonly used for response data, which are supported by **mirt**
- Estimating single and multiple group IRT models with marginal maximum likelihood estimation in **mirt**
- Person, item, and model fit statistics
- Multiple-group estimation, DIF and DTF

References

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

Cai, L. (2010). High-Dimensional exploratory item factor analysis by a Metropolis-Hastings Robbins-Monro algorithm. *Psychometrika, 75*, 33-57.

Chalmers, R. P. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software, 48*, 1-29.

Chalmers, R. P., Counsell, A., and Flora, D. B. (in press). It might not make a big DIF: Improved Differential Test Functioning statistics that account for sampling variability. *Educational and Psychological Measurement*.

Gibbons, R. D., Darrell, R. B., Hedeker, D., ... (2007). Full-Information item bifactor analysis of graded response data. *Applied Psychological Measurement, 31*, 4-19.

Maydeu-Olivares, A., Hernández, A., & McDonald, R. P. (2006). A Multidimensional Ideal Point Item Response Theory Model for Binary Data. *Multivariate Behavioral Research, 41*, 445-471.

Suh, Y. & Bolt, D. (2010). Nested logit models for multiple-choice item response data. *Psychometrika, 75*, 454-473.