Item Response Theory in R

Thomas Rusch

WU Vienna

July 17, 2017
1. Item Response Theory
2. Unidimensional IRT
3. Multidimensional IRT
4. Multiple Group IRT, DIF, and DTF
5. Conclusion
Overview

Part 2: Item Response Theory

Primarily about possibilities in R for applying IRT models. We focus on model fitting, model checking, item/person parameter extraction and diagnosing item sets.

---


We use a non-comprehensive list of packages for which there is an associated peer-reviewed publication (skipping some great packages).

- Overview of basic and advanced IRT models useful for applied item analyses
- Hands-on applied analysis with various packages for fitting item response models, checking model fit and item diagnostic utilities, estimation of latent trait values
- Utilizing models to detect non-standard ICC or differential item functioning or latent groups

1Big thanks to Phil Chalmers for letting me use mirt workshop material
Item response theory (IRT) is a set of latent variable techniques specifically designed to model the interaction between a participant's ability, or latent trait, with item level stimuli (difficulty, guessing, etc.).

Three main reasons to use IRT:

- **Model a set of items** (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 set of items** to obtain estimates of the latent trait(s) for individual participants
Understanding our data

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

```markdown
## Item_1 Item_2 Item_3 Item_4 Item_5 Item_6
## [1,] 0 1 0 2 0 1
## [2,] 1 1 0 2 0 2
## [3,] 0 1 0 2 1 1
## [4,] 0 1 0 1 0 0
## [5,] 0 1 1 0 1 1
## [6,] 1 1 1 2 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 $\alpha$, McDonald’s $\omega$, 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

Rasch Models
A popular class of IRT models were developed to model how a subject’s ability ($\theta_n$) was related to answering a test item $i$ ($0 = \text{incorrect}, 1 = \text{correct}$) given a single item-level property (difficulty), and how this could be understood probabilistically.

\[
P(y_i = 1|\theta_n, b_i) = \frac{\exp(\theta_n - b_i)}{1 + \exp(\theta_n - b_i)}
\]

- This is the \textit{Rasch model}
- Given some ability, $\theta_n$, the probability of positive endorsement is non-linearly related to only the item difficulty ($b$, location parameter).
  In canonical form: $\log(P/(1 - P)) = \theta - b$
- This model can be reparametrized so that $\theta + d = \theta - b$ (we prefer easiness to difficulty).
Figure 1: A dichotomous IRT model is similar to a logistic regression model; however, in IRT $\theta$ is not observed directly.
Rasch models (dichotomous) - II

- This is a rather strict model and not well suited for modelling
- It is important because of its special properties:
  - Sum of 1's (raw score) is sufficient and fair for comparing people
  - Allows “specific objective” comparisons
  - Item parameters can be estimated by conditioning out the person parameter (Conditional Maximum Likelihood - CLM)
Rasch models (dichotomous) - III

**R functionality:**

- **eRm**: Fit by CML (`RM()`)  
- **ltm**: Fit by Marginal ML (`rasch()`)  
- **mirt**: Fit by various estimation (stochastic MML or MH-RM, EM) `mirt(..., itemtype="Rasch")`  
- **plRasch**: Fit by pseudo-likelihood loglinear models (`RaschPLE()`)
Rasch models (polytomous) - I

Rasch models can be extended to polytomous answers. These Rasch-type models retain the properties of the Rasch model (only location parameters for the item, sufficiency, specific objectivity, CML).

One general expression of these models is:

\[
P(y_{ni} = k|\theta, \psi) = \frac{\exp(ak_k\theta_n + d_{ik})}{\sum_{j=1}^{k_i} \exp(ak_j\theta_n + d_{ij})}.
\]

For the Partial Credit Model (PCM) the \(ak_k\) values are treated as fixed and ordered from 0 to \((k_i - 1)\). This indicates that each successive category is scored equally (the \(ak_k\) values are often interpreted as scoring coefficients). We have one parameter for each item \(\times\) category.

- **eRm**: PCM(), in **mirt**: mirt(..., itemtype="Rasch")
Rasch models (polytomous) - II

The PCM can be further generalized by estimating the $a_k$ values directly, which then indicates the ordering of the categories empirically. This is called the Nominal Response Model (NRM).

- **mirt**: `mirt(..., itemtype="nominal")`

For the Rating Scale Model (RSM) we have a PCM where all items have the same number of the categories and constant scoring of categories over items, only the location of items differs, so

$$\theta_n + d_{ik} = \theta_n + d_i + c_k$$

This boils down to modelling

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

as the difference between adjacent Rasch models (dichotomizing the item at each category, and estimating separate Rasch models).

- **eRm**: RSM, in **mirt**, **ltm**: Dichotomizing strategy.
Restricted Rasch-type models

There is a class of Rasch models that arise from decomposing the item/person parameters to $p$ basic parameters $\eta$,

$$d_{ik} = \sum_{l=1}^{p} w_{ikl} \eta_l$$

The matrix of $w_{ikl}$ is called Q-Matrix.

- **LLTM**: Linear decomposition of the $d_i$ in the Rasch Model
- **LRSM**: Linear decomposition of the $d_i$ in the Rating Scale Model
- **LPCM**: Linear decomposition of the $d_{ik}$ of the Partial Credit Model

These models can also be used to measure change over time (usually in the person parameter). For this, each item at each time point after baseline is seen as its own *virtual* item. Each virtual item parameter can then be expressed as the parameter at baseline plus a change. Since we are having Rasch models a change of item difficulty is in effect the same as change of person ability. In **eRm** one uses the option `mpoints` for measurement points. It is also possible to include group information (and covariates) as `groupvec`.
Important Functions in eRm

- RM(data), PCM(data), RSM(data), LTM(data), LPCM(data), LRSM(data): Model fitting functions
- summary(obj), residuals(obj), coef(obj), anova(obj): Generics
- gofIRT(obj), LRtest(obj), Waldtest(obj), NPtest(obj): Test and Statistics for Model fit
- plotICC(obj), plotGOF(obj), plotDIF(obj), plotPImap(obj), plotPWmap(obj): Plot functions
- person.parameter(obj): Extract person parameters
- item_info(obj), test_info(obj): item or scale information
- person.fit(obj), item.fit(obj): person and item fit statistics

Note: Not all functions are necessarily working with all object classes.
**eRm**

A look at functionality of **eRm**.
Explanatory Rasch Models - I

The idea of reparametrization used in the restricted Rasch models can be generalized to any functional of person- and item parameter and respective covariates to explain scale/item variation.

This is useful because often we want to know the effect of including additional information to help explain the observed response patterns.

For this, we can reformulate the models as from the class of Generalized Linear Mixed Models (or GLLAMM - generalized linear latent and mixed model).

Mixed models (aka multilevel models, hierarchical models) consist of:

- **Fixed effects**: Fixed effects are exhaustive and by design. They are related to estimating mean effects.
- **Random effects**: Effects for which we make a distributional assumption and the concrete values are only realizations. They are related to estimating variance (components).

This formulation makes estimation of complex models feasible.
Mixed Effect Rasch Model

In the case of the Rasch model, the $d$ may be seen as the fixed effects and the $\theta$ are random effects (drawn from a population) leading to a random intercept model. This changes the formulation slightly but with profound effects for the applicability:

$$P(y_i = 1|d_i, \theta_n) = \frac{\exp(\zeta_{ni})}{1 + \exp(\zeta_{ni})} = \frac{\exp(\theta_n + d_i)}{1 + \exp(\theta_n + d_i)}$$

with $\theta \sim F_n(\psi)$, usually $F_n(\psi)$ is the multivariate normal distribution with mean vector 0 and Variance-Covariance matrix $\Sigma$. 
In this framework we can expand the linear predictor $\zeta$ (vector form) to\(^2\).

$$\zeta = W\eta + X\beta + Zr + \epsilon$$

with $W$ a fixed effect design matrix for the $d$, reparametrising them with $\eta$ (e.g., as in the extended Rasch models) and the $X$ a fixed effect design matrix for the $\theta$ reparametrizing them to $\beta$, $Z$ being a random effect design matrix of the random effects $r$ specifying item- or person-specific random effects and $\epsilon$ being a random effect error terms (we usually assume the random effects to be normal).

An example for $X$ would be treatment/control group and gender indicated by 0s and 1s, or continuous variables such as age. An example for $Z$ would be an additional random effect for the school.

\(^2\)We now skip the indices
Now this is a very general formulation but many models can be seen as special cases of this and estimated in a single framework, including

- The restricted Rasch models and or Rasch models with extension by item specific covariates (using $W$ and $Z$)
- Latent regression models where the $\theta$ is regressed on covariates (using $X$ and $Z$)
- Person-by-item covariate Rasch models (using $W$ and $X$)
- Multidimensional models (using partitioned version of $X$)
- Random and fixed effects on person and item level (using $W$, $X$ and $Z$)
- All combinations of the above
For the Rasch model extensions discussed above, one can use the generalized linear mixed effect model function `glmer()` from `lme4`. This is an incredibly flexible way to do Rasch (or 1PL) type modelling by the formula interface.

The syntax is essentially

```
glmer(responses ~ 0 + fixed effects + (random effect | nesting))
```

Generics for the models exist. Drawback is that one needs to know how to model this properly, which is quite involved (setting up $W$, $Z$ and $X$).
**lme4**

A look at functionality of **lme4**.
Exercise: Rasch Modelling in R

Exercises pertaining to Rasch Modelling are available in Exercise_03.html.
Unidimensional IRT

Parametric IRT
Parametric IRT models (dichotomous) - I

IRT models generalize the approach also taken in the Rasch model by allowing for a more flexible specification of relationship between ability and item answer. They are better suited for modelling but their results usually lack the Rasch properties.

- The most famous IRT model is the *Two Parameter Logistic Model (2PL)*\(^3\)

\[
P(y = 1|\theta, a, d) = \frac{1}{1 + \exp(-(a\theta + d))}
\]

In canonical form: \(\log(P/(1 - P)) = a\theta + d\)

This adds a second item parameter \(a\) to the Rasch model, the item discrimination or slope of the ICC. The Rasch model is realized when the slope parameters are fixed to a constant (usually 1).

\(^3\)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 \(b = -d/a\)
Figure 2: Item response curves when varying the slope and intercept parameters in the 2PL model
Parametric IRT models (dichotomous) - II

Generalization of the 2PL model are also possible to accommodate for other common testing phenomenon, such as guessing (3PL) and/or careless responding effects. The most general one in use is the *Four Parameter Logistic Model (4PL)*

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

which when specific constraints are applied reduces to the 3PL, 2PL, and Rasch model.

- Given \(\theta\) 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)
Figure 3: Item response curves when varying the lower and upper bound parameters in the 4PL model
IRT in R

R functionality:

- **ltm**: 2PL `ltm()` and 3PL (lower asymptote) `tpm()`; fit by Marginal ML
- **mirt**: 2PL `mirt(..., itemtype="2PL")`, 3PL (with lower `itemtype="3PL"` or upper "3PLu" asymptote), 4PL `itemtype="4PL"`; fit by various estimation (stochastic MML or MH-RM, EM)
- **MCMCpack**: 2PL `MCMCirt1d`
To help understand how the parameterizations in IRT models affect the shape of the probability response curves, mirt has 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:

```r
library('mirt')
itemplot(shiny = TRUE)
```
Polytomous item response models exist for ordinal categories, rating scales, partial credit scoring, unordered categories, and so on. They usually generalize the models we already mentioned in the Rasch context.

*Graded Response Model*: Good for Likert items which are often modeled by ordinal/rating scale. We can express it simply as the difference:

\[
P(y = k | \theta, \phi) = P(y \geq k) - P(y \geq k + 1),
\]

between adjacent 2PL models (dichotomizing the item at each category, and estimating separate 2PL models). This generalizes the Rating Scale Model to include different discrimination for the items. This strategy can of course be extended to the 3PL and 4PL.
Parametric IRT models (polytomous)

There are also models from the so-called ‘divide by total’ family of IRT parameterizations, such as the Generalized Partial Credit Model and Generalized Nominal Response Model.

They are of following form

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

Note the \(a\), which means the discrimination effect for each person can be different. The \(ak_k\) values indicate the ordering of the categories.

- For the generalized partial credit model the \(ak_k\) values are treated as fixed and ordered from 0 to \((k - 1)\) as in the PCM.
- In nominal models, some \(ak_k\) values are estimated to indicate the ordering of the categories empirically.

If \(a = 1\) the Rasch-type models result. I am not aware of 3 or 4 PL versions.
**Unidimensional plots (polytomous)**

**Figure 4:** Probability curves for ordinal/graded (left), generalized partial credit (center), and nominal (right) response models
Important Functions in `ltm`

- `rasch(data), ltm(data), tpm(data), grm(data), gpcm(data)`: Model fitting functions (with constraints)
- `GoF(obj), margins(obj), unidimtest(obj)`: Tests for Model fit
- `plot(obj), summary(obj), residuals(obj), coef(obj), anova(obj)`: Generics
- `factor.scores(obj)`: Extract person parameters
- `information(obj)`: item or scale information
- `person.fit(obj), item.fit(obj)`: person and item fit statistics

Note: Not all functions are necessarily working with all object classes.
**ltm**

A look at functionality of **ltm**.
Important Functions in `mirt`

- `mirt(data, model, itemtype)`, `bfactor(data)`, `mixedmirt(data, model, formula)`: Model fitting functions for different item types
- `wald(obj)`: Test and Statistics for Model fit
- `itemplot(obj, options)`: Plot functions (many different options)
- `plot(obj), summary(obj), fitted(obj), residuals(obj), coef(obj), anova(obj)`: Generics
- `fscores(obj)`: Extract person parameters
- `iteminfo(obj), testinfo(obj), expected.item(obj)`: item or scale information
- `M2(obj), itemfit(obj)`: person and item fit statistics
**mirt**

A look at functionality of **mirt**.
Exercise: Unidimensional Parametric IRT Modelling in R

Exercises pertaining to parametric IRT are available in `Exercise_04.html`. 
Nonparametric IRT
IRF in parametric IRT

So far we looked at parametric IRT, meaning that we restricted the functional form of the RF (response function for item or category).

Typical RF assumptions:

- **Monotonic S-Curves:**
  - Logistic Models (Rasch, IRT), Probit Models, More Obscure Links (Log-Log, Cauchit)
  - By far the most popular

- **Unimodal non-monotonic concave RF:**
  - Deviation from location causes a decrease in the response probability in either way
  - Middle categories in polytomous models
  - For dichotomous items: *Ideal point models* have the form

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

- May be with non-ability based measures (e.g., personality traits, preference ratings, etc).
- For example, I think R is ok. You may disagree because you think R is not ok, or because you think R is super.
- Can be fit with mirt(..,itemtype="ideal")
Nonparametric IRT - I

Parametric IRT can thus be restrictive or needs to be justified (e.g., why use square in the ideal point model?)

Non- or semiparametric aproaches allow for any RF.

The RF needs to be estimated though.

Unless the parametric approach is needed on theoretical grounds, I’d always first explore with nonparametric models. This also emphasizes exploration and graphical presentation.
The nonparametric way is to estimate the RF directly from the data. The *Nonparametric IRT Model* is therefore fairly general

\[ P(y_i = k|\theta) = f_{ki}(\theta). \]

This means every item \( i \)'s category \( k \) can have as RF its own functional form \( f_{ki} \) which we estimate by \( \hat{f}_{ki} \). This is called non-parametric because one has an infinite number of possible functional forms.

How to find \( \hat{f}_{ki} \)?

- **Smoothing**: From the observed \( y_i \) the \( \hat{f}_{ki} \) is estimated by smoothing (splines or kernel smoothing).
- **Estimate step functions**: \( \hat{f}_{ki} \) is approximated by step functions.
Estimating RF by Smoothing

There are two main ways of smoothing to estimate the RF:

- **Spline Response Models**: They estimate $\hat{f}_{ki}(\theta)$ via smoothing splines. A $B$–spline function is a piecewise polynomial function of degree $< g$ as a function of $\theta$. The points where the splines meet are called knots. The overall spline is the weighted sum of the local polynomials (basis functions) over the knots. For cubic basis functions, it is called a n-spline. They can be fitted in `mirt` with `mirt(..., itemtype=\"spline\")`.

- **Kernel Smoothing Models**: They estimate $\hat{f}_{ki}(\theta)$ via Kernel smoothing. The $y$ are smoothed via a Kernel function of $\theta$. A Kernel is a sort of distance window over which to smooth with weight corresponding to the Kernel function value. There are many Kernels, e.g., “uniform”, “Gaussian“. They can be fitted with the `ksIRT()` from `KernSmoothIRT`.

In practice, the two approaches need not differ much. Splines may be better with small data sets and sharp turns in the IRF. The Kernel approach is easier to analyze mathematically and extends straightforwardly to multiple dimensions.
Important Functions in **KernSmoothIRT**

- **ksIRT(data,key,weights)**: Model fitting function. *format* specifies whether multiple choice (1), rating scale or partial credit (2) or nominal treatment (3) or allows mixed formats. Other scoring rules can be specified with *weights*.
- **plot(obj,plottype)**: Plot functions (many different options)
- **subjthetaML(obj)**: Extract person parameters
- **subjEIS(obj)**, **subjETS(obj)**, **subjscoreML(obj)**, **subjOCC(obj)**: subject-wise scores (expected item and test, ML scores; OCC gives all incl. observed)
**KernSmoothIRT**

A look at functionality of **KernSmoothIRT**.
Another nonparametric approach is Mokken scaling

- Item Response Function assumed is a monotonic non-decreasing function (possibly non smooth)
- Works with dichotomous and polytomous items
- Allows to partition data according to nonparametric models that imply ordinality

Very useful as a diagnostic suite for checking assumptions in parametric and nonparametric IRT.
The underlying models arise from the assumptions:

1. Unidimensionality of $\theta$
2. Local Independence
3. Latent Monotonicity
4. Nonintersection (RF do not intersect)

The two underlying models are:

- *Monotone homogeneity model (MHM)* or *nonparametric graded response model* (fulfills 1-3)
- *Double monotonicity model (DMM)* (fulfills 1-4)

Rasch-type models are a special case of DMM, 2-4PL of MHM, which can be used for checking.
Main functionalities in package **mokken**:

- **Scalability coefficients** $H_{ij}$: Covariance over maximal covariance given the marginal per item pair. If they belong on the same Mokken scale, they should have a positive scalability. ($\text{coefH}()$)
- **Item Scalability** $H_i$: Covariance between item and item set score minus the item over the maximum covariance given the marginals. If the item belongs to that set on the same Mokken scale, a positive item scalability $>0.3$ is needed ($\text{coefH}()$)
- **Test scalability** $H$: For a set of items, $H_i$ over all items on a set. If $H = 1$ it is a Guttmann scale. Weak scale if $H < 0.3$ ($\text{coefH}()$)
- **Algorithms** that partition a set of item to scales meeting the Mokken criteria and a set of unscalable items. Function `aisp()` (use `search="ga"`, the item’s number indicates the scale, 0 is unscalable).
- **Diagnostic assumption checks**
Checking Assumptions with Mokken Scaling

- **Stochastic ordering of latent trait (SOL):** Is essentially the MHM (Ass. 1-3) and for it to hold it means that $0 \leq H_{ij} \leq 1$ for all $i \neq j$ and $0 \leq H_j \leq 1$ for all $j$ and $0 \leq H \leq 1$. Necessary to use the score to rank (groups of) subjects (strong SOL does not hold for polytomous MHM).

- **check.monotonicity():** Checks Assumption 3. The `summary()` gives with `#vi` the number of violations, the sum of those greater than a cut-off and some standardized versions (`sum/#ac`) as well as a significance test (`#zsig`). Values $>0$ are warning signs.

- **check.pmatrix(), check.restscore():** Check Assumption 4. The `summary()` gives the `#vi`, the sum and `sum/#ac`. Values $>0$ are warning signs.

- **check.iio():** Invariant item ordering (IIO) The `summary()` gives `#vi`, the sum and a significance test (`tsig`). Values $>0$ are warning signs. One can also use `backward.selection()` to see where the violations occurred, what happened after they were removed. One lands at a scale with IIO. One can also use `plot()`.
Important Functions in **mokken**

- `coefH(data)`: Scalability coefficients
- `aisp(data)`: Scale partitioning
- `check.iio(data), check.pmatrix(data), check.restscore(data), check.monotonicity(data)`: Assumption checks
**mokken**

A look at functionality of **mokken**.
Exercise: Nonparametric Unidimensional IRT Modelling in R

Exercises pertaining to nonparametric IRT are available in Exercise_05.html.
Multidimensional Item Response Theory

Multidimensional IRT
We now turn to multidimensional IRT (MIRT).

Luckily, there is little new, we simply extend some of the models we looked at before.

Our canonical workhorse will be mirt(). Nearly all we did with mirt so far extends straightforwardly to two dimensions.
Multidimensional IRT models

Multidimensional IRT (MIRT) models replace the single $\theta$ and $a$ values with vectors $\theta$ and $a$, respectively. This means we now fit a surface. This is analogous to the transition from zero-order logistic regression to multiple logistic regression.

The Multidimensional 4PL Model is

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

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

- Similar relationship exists for the Multidimensional 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.

The *Multidimensional GPCM, Multidimensional GNRM* are

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

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).
Figure 5: Probability curves for multidimensional 2PL and ordinal (top), generalized partial credit and nominal models (bottom)
The multidimensional models in the previous graphs are know as 'compensatory' models. The reason for this is that

- A low $\theta_k$ parameter on one of the 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 $\theta$ since they are constructed by multiplying individual response curves, e.g., for the 2PL:

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

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

- *MIRT models* can be understood as non-linear extensions of more traditional factor analysis methodology, and as such have **exploratory** and **confirmatory** aspects.

- For exploratory models, the orientation of the $\theta$ 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).
Fitting exploratory multidimensional models with \texttt{mirt} is very straightforward:

One fits the same model as we did in the unidimensional case (via argument \texttt{itemtype}) but now also specifies the \texttt{model}, which in this case is simply an integer with the number of dimensions \texttt{dim}.

The call is therefore \texttt{mirt(data,model=dim,itemtype="irtmodel")}.

Other notable packages for multidimensional IRT are \texttt{plRasch} for Rasch models and \texttt{ltm} which can also do some multidimensional IRT.
Confirmatory MIRT in R

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

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
Latent regression for IRT models is possible in `mirt` with `mixedmirt()`.

The latent regression model attempts to decompose the $\theta$ effects into fixed-effect components to *explain* differences between individuals in different populations.

If the IRT models are not from the Rasch family, various item-specific slopes must be fixed to a constant in order to identify the $\Sigma$ elements.
Purpose of mixed-effects modeling is to include continuous or categorical item and person predictors into the model directly by way of the intercept parameters. An example of including a fixed effect predictor into the model at the person level would be the inclusion of ‘Gender’, where an indicator coding is used to change the expected probability to:

\[
P(x = 1; \theta, \Psi, \beta_{male}) = g + \frac{(u - g)}{1 + \exp[-(a'\theta + d + \beta_{male}X)]}.
\]

Notice here that \( \theta \) is not directly decomposed and instead additional intercepts are modeled. This model is distinct from the latent regression model, which has the form

\[
P(x = 1; \theta, \Psi, \beta_{male}) = g + \frac{(u - g)}{1 + \exp[-(a'\theta + d)]}.
\]

where \( \theta = \beta_{male}X + \epsilon \).
Mixed IRT in R

mixedmirt() in the mirt package was designed to include fixed and random intercepts coefficients into the modeling framework directly.

For the M4PL model,

$$P(Y) = g + \frac{(u - g)}{1 + \exp(-(W\eta + X\beta + Zr + \epsilon))}$$

where similar terms are the same as in the mixed effects Rasch models: $W\eta$ the item fixed effects, $X\beta$ the person fixed effects (latent regression) and the $Zr$ controls the person- and/or item-specific random effects terms and $\epsilon$ a random error.

Effects organized to explain person effects (i.e., latent regression models), but also can explain variability in the test itself (i.e., explain why some items are more difficult than others, account for speeded effects,). Hence, the above model can simultaneously capture item and person effects.

Generalizations of this equation are fairly simple to polytomous responses, however intercept designs require slightly more care.
mixedmirt() syntax

Has the usual data and model arguments as before, however it also supports

- covdata – a data.frame of person-level covariate information, and
- itemdesign – a data.frame of item design based effects

The fixed effects are controlled with

- fixed – a standard R formula to decompose the intercept effects (e.g., ~ items + gender*IQ). Corresponds to the $W\eta$ effects
- lr.fixed – an R formula to decompose the latent trait(s). Corresponds to the $X\beta$ effects
Random effect terms have a syntax similar to the `nlme` package:

- `random` – an | separated R formula indicating random intercepts and slope effects for the intercepts (e.g., ~ 1 | group) (Zr)
- `lr.random` – an | separated R formula indicating random intercepts and slope effects for the trait (Zr)

By modeling variability with the random effects, fixed effects predictors can be used to ‘explain’ different sources of variation (e.g., why schools may be different, or why some items are more difficult than others). Hence, random effects are generally interpreted as ‘residual variation’.
**mirt**

A look at functionality of **mirt**.
From the Rasch family there is a multidimensional model to measure change over time, the *Linear Logistic Model with Relaxed Assumption (LLRA)*

- Keeps the Rasch properties of specific objectivity for the time change
- Only allows to estimate the change but not the baseline (condition on those)
- In the most general formulation specifies each item as its own dimension
- Relaxed assumptions does not mean little assumptions (crucial: changes must be independent of each other, including changes of the same person for other items)
Measuring Change - II

The model at $T_1$ (baseline) is

$$P(X_{vih1} = 1|T_1) = \frac{\exp(h\theta_{vi} + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta_{vi} + \omega_{il})}$$

At any measurement point $T_t$ ($t \in \{2, 3, \ldots \}$)

$$P(X_{vih} = 1|T_t) = \frac{\exp(h\theta_{vit} + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta_{vit} + \omega_{il})} = \frac{\exp(h(\theta_{vi} + \delta_{vit}) + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l(\theta_{vi} + \delta_{vit}) + \omega_{il})}$$

$\delta_{vit} = \theta_{vit} - \theta_{vi1}$ ... amount of change of person $v$ for trait $i$ between time $T_1$ and $T_t$

$h$ ... $h$-th response category ($h = 0, \ldots, m_i$)

$\omega_{ih}$ ... parameter for category $h$ for item $i$
Model Reparametrization

The flexibility of LLRA now arises from a (linear) reparametrization of $\delta_{vit}$ to include different effects:

$$\delta_{vit} = w_{it}^T \eta$$

$w_{it}^T$ ... row of design matrix $W$ for item/trait $i$ up to $T_t$
$\eta$ is a vector of parameters typically describing effects for factor or covariate levels like groups as well as trend and interactions. Often it will look like this:

$$\delta_{vit} = \sum_j q_{vjit} \lambda_{jit} + \tau_{it} + \sum_{j<l} q_{vjit} q_{vilit} \rho_{jlit}$$

$q_{vjit}$ ... dosage of covariate/factor $j$ for trait $i$ between $T_1$ and $T_t$
$\lambda_{jit}$ ... main effect of the covariate/factor $j$ on trait $i$ at $T_t$
$\tau_{it}$ ... trend effect on trait $i$ for $T_t$
$\rho_{jlit}$ ... interaction effects of treatments $j$ and $l$ on trait $i$ at $T_t$
The **eRm** package offers functionality for LLRA with the `LLRA()` function. This function fits the quasi-staurated model (each item is its own dimensions).

- Generics as for all `eRm()` models.
- `plotTR()` and `plotGR()` to plot.
- `collapse_W()` convenience function specify Q-Matrix to fit restricted models (i.e., restrict treatment or trend effects over items, groups and subjects)
**eRm**

A look at functionality of **eRm**.
Exercise: Multidimensional IRT Modelling in R

Exercises pertaining to multidimensional IRT are available in Exercise_06.html.
Multiple Groups

Multiple Group IRT
Multiple Group models

*Multiple group analysis (MGA)* takes into account empirical groups 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. We discussed this in parts already from the perspective of reparametrization of item and person parameters in Explanatory IRT. Here we return to this in a more “canned” and specialized versions, and also extend it.

MGA has two principle approaches:

- **Hard partitioning**: Completely separate the data according to membership (e.g., multiple single groups, reparametrization) up to a single group model (i.e., ignore group membership)
- **Soft partitioning**: Models between these extremes, where we try to find a model that does not need strict independence while being mindful of differences (e.g., partial parameter restriction, finite mixture models)
Multiple Group Likelihood

The log-likelihood contribution for an individual that is evaluated for these models is

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

The weights $w_{Gg}$ are $\in [0; 1]$. If $w_{Gg} = \{0, 1\}$ we have hard partitioning. Parameters can therefore be constrained to be equal across members of different groups, or freely estimated, and allows for nested model comparisons.

We distinguish by focus the following the special cases:

- MGA to look at *measurement invariance*, where items behave the same depending on the group
- MGA as *differential item functioning (DIF)*, where items function differently depending on the group
For a *hypothesis-driven, confirmatory* approach to invariance, we can use the `multipleGroup()` function from `mirt`. It defaults to the completely independent groups approach.

Because of the property that the restricted model is nested in the unrestricted one and can simply use `anova()`.

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

\[ P_{\text{focal}}(k = K|\theta) \neq P_{\text{reference}}(k = K|\theta) \]

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 \( \theta \) metrics during estimation.
- Different types of DIF exist, but largely these have been grouped into uniform and non-uniform DIF.
  - In R: `difR`, `mirt`, `psychotree`
Differential item functioning in R - I

With \texttt{difR}: For uniform (u) and non-uniform (nu) DIF. Only for dichotomous item models.

- 11 DIF methods, including non-IRT methods
- IRT methods:
  - \texttt{difLord()}, \texttt{difGenLord}() (u and nu DIF in two and multiple groups)
  - \texttt{difRaju}() (un and nu DIF for two groups)
  - \texttt{difLRT}() (u DIF in Rasch models)
  - \texttt{dichoDif()}, \texttt{genDichoDif}(): Convenience function to check DIF (all methods for 2 or more groups)
**difR**

A look at functionality of **difR**.
Differential item functioning in R - II

With **mirt**: Via the Wald test or the likelihood-ratio test through the automated testing function `DIF()`.  

- `DIF(obj)` requires a small pre-selected number of invariant “anchor” items, and that the group hyper-parameters are freed for all but one group. This ‘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
**mirt**

A look at functionality of **mirt**.
Invariance and DIF - Exploration

It may be that we have a possibly large number of groups and for some of them invariance holds but we don’t know which ones. One can therefore automatically look for groups for which there is no invariance.

One package that allows to do that is **psychotree** which starts from a single group model and greedily partitions the data along groups with parameter instability (defined by categorical or metric covariates). This automatically detects groups for which invariance does not hold.

It currently works with Rasch models

- `raschtree()`: Model-based recursive partitioning of Rasch Model
- `pctree()`: Model-based recursive partitioning of PCM
- `rstree()`: Model-based recursive partitioning of RSM
**psychotree**

A look at functionality of **psychotree**.
Mixture IRT models

What if we do not want or cannot have hard partitioning? Assign each observation to a number $g < \infty$ of $G$ latent groups with a certain weight from $[0; 1]$. Each group has different parameters. This is called a *finite mixture model*:

$$P(y) = \sum_{g}^{G} \pi_g P_{IRT}(y; \psi_g)$$

with $\pi_g$ being the mixing proportion ($\sum_{g}^{G} \pi_g = 1$) and $P_{IRT}(x; \psi_g)$ being the IRT model in each group $g$ with $\xi_g$ all of its parameters. Concomitant variables $z$ additionally allow to model the $\pi_g := \pi(g) = h(z)$ (e.g., with logistic model).

These models allow to:

- Detect and explain clusters of observations
- Detect (artifical) DIF between clusters
- Incorporate additional (unexplained) heterogeneity
- Connect IRT to Latent Class Models
Currently, only Rasch type models are available for mixture models.

- **mRm**: `mrm(data, cl=groups)` Mixed dichotomous Rasch models
- **psychomix**: `raschmix(formula, data, k=groups)` Mixed dichotomous Rasch models with possible concomitant variables
- **mixRasch**: `mixRasch(data, n.c=groups)` Dichotomous and polytomous Rasch models.

All feature some standard generics.
**mixRasch** and **psychomix**

A look at functionality of **mixRasch** and **psychomix**.
Exercise: MGA, Measurement Invariance and DIF in R

Exercises pertaining to MGA, MI, and DIF are available in Exercise_07.html.
Conclusion

We made a tour through the availability of IRT in R and it is impressive what we can already do.

But there is even much more going on in R, incl. support for

- Customized IRT models and latent densities in **mirt**
- Additional IRT models (testlet, NOHARM, Rasch copula, facets, ISOP, LSEM) in **sirt**
- Item calibration with **plink**
- Test equating with **kequate**
- Computerized adaptive testing with **catR**
- Discrete latent traits (DINO, DINA, etc.) models with **CDM**, **GDINA**, **NPCD**

Unfortunately we were forced to disregard those important areas and great software. A big shout-out to the authors of these packages.

What is even better: The development continues. With what we did today you should find your way around the growing ecosystem of IRT in R (and perhaps even develop!)
End of Workshop

This is the end of our workshop (Yay!). Just to review what we learned:

- R as a software for data analysis
- Basic concepts of IRT (item response function, etc)
- Many different IRT models (unidim, multidim, parametric, nonparametric, mixed, etc)
- R packages for IRT analysis
- Estimating single and multiple group IRT models in R
- Multiple-group estimation, DIF, DTF and Mixtures


References - III


Please attribute **Thomas Rusch**. This work is licensed under CC-BY-SA:

https://creativecommons.org/licenses/by-sa/4.0/