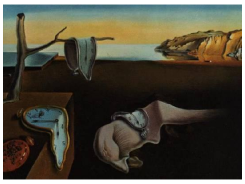

Journal of Statistical Software

May 2012, Volume 48, Code Snippet 1. http://www.jstatsoft.org/

IRTrees: Tree-Based Item Response Models of the GLMM Family

Paul De Boeck
University of Amsterdam

Ivailo Partchev
Cito Arnhem

Abstract

A category of item response models is presented with two defining features: they all (i) have a tree representation, and (ii) are members of the family of generalized linear mixed models (GLMM). Because the models are based on trees, they are denoted as IRTree models. The GLMM nature of the models implies that they can all be estimated with the `glmer` function of the **lme4** package in R. The aim of the article is to present four subcategories of models, the first two of which are based on a tree representation for response categories: 1. linear response tree models (e.g., missing response models), 2. nested response tree models (e.g., models for parallel observations regarding item responses such as agreement and certainty), while the last two are based on a tree representation for latent variables: 3. linear latent-variable tree models (e.g., models for change processes), and 4. nested latent-variable tree models (e.g., bi-factor models). The use of the `glmer` function is illustrated for all four subcategories. Simulated example data sets and two service functions useful in preparing the data for IRTree modeling with `glmer` are provided in the form of an R package, **irtrees**. For all four subcategories also a real data application is discussed.

Keywords: generalized linear mixed models, item response models, multidimensional.

1. Introduction

IRTree models are models for item response data with a tree structure. The tree structure can apply to (1) the response categories of items (response trees) or (2) the latent variables underlying the item responses (latent-variable trees), or to both. The general motivation for these models is twofold: IRTree models are rather flexible and informative so that they can deal with issues that often remain unresolved with other approaches, and IRTree models are members of the generalized linear mixed model (GLMM) family so that general purpose software is available. The software motivation refers to the **lme4** package (Bates *et al.* 2011) in

R (R Development Core Team 2012), which is a user friendly and fast kind of general-purpose software for GLMMs.

Because the IRTree models to be discussed are GLMMs, they are not new models. The novelty lies in a general tree framework and in the (re)formulation of a set of IRT models as GLMMs. The IRTree formulation opens perspectives on interesting extensions and on the use of a general purpose kind of GLMM software. On the one hand, it is a clear benefit that a model can be reformulated as a GLMM, but on the other hand GLMM models come with the limitation that bilinear functions cannot be dealt with. Especially from an item response theory point of view this is a substantial restriction, since all weights need to be fixed and therefore item discriminations cannot be estimated. The two-parameter model and the three-parameter models are therefore excluded. In some fields of data analysis this is not a serious limitation. For example, the one-parameter assumption of compound symmetry is shared with ANOVA, and for item response growth curve models (Hung 2010; McArdle *et al.* 2009) the one-parameter model is a common model.

Section 2 discusses response trees, and Section 3 is dedicated to latent-variable trees. Within each section, there are four subsections: (1) motivation, (2) model formulation, (3) data preparation and data analysis based on simulated data, and (4) application to real data. The example data sets and two service functions useful in preparing the data for IRTree modeling with **lme4** are provided in the form of the R package **irtrees** which is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=irtrees`.

2. Response tree models

2.1. Why response tree models?

Item response models are models for categorical item responses. The response categories are dichotomous (yes vs. no, correct vs. incorrect) or polytomous (e.g., multiple choice), and in case of the latter, the categories are commonly either nominal or ordinal. The three well-known models for binary items are the Rasch model, the two-parameter and three-parameter models (Hambleton *et al.* 1991). Multiple categories are often collapsed into binary categories, such as correct vs. incorrect. Without such a reduction, the most familiar available models for polytomous item responses are Bock's model for nominal categories (Bock 1972) or variations thereof, and two models for ordered categories: the partial credit model (Masters 1982) and the graded-response model (Samejima 1969) and extensions thereof. The response tree models to be discussed here are inspired by a third such model, the sequential model (Tutz 1990).

Response tree models are models for categorical data such that (1) the response categories can be represented with a binary response tree, and (2) the response process can be interpreted as a sequential process of going through the tree to its end nodes. The response trees can be linear or nested. When linear, one branch from each internal node leads directly to an end node (a response category) while the other leads to the next internal node. See the left part of Figure 1 for an example with as response categories "no", "perhaps", and "yes", and with internal nodes Y_1^* and Y_2^*. When nested without being linear, both branches from the higher-ordered internal nodes lead to a lower-ordered internal node and only indirectly to an end node (response category). See the right part of Figure 1 for an example with Y_1^*, Y_2^*, and Y_3^* as internal nodes for analogy items with four response categories to be explained later.

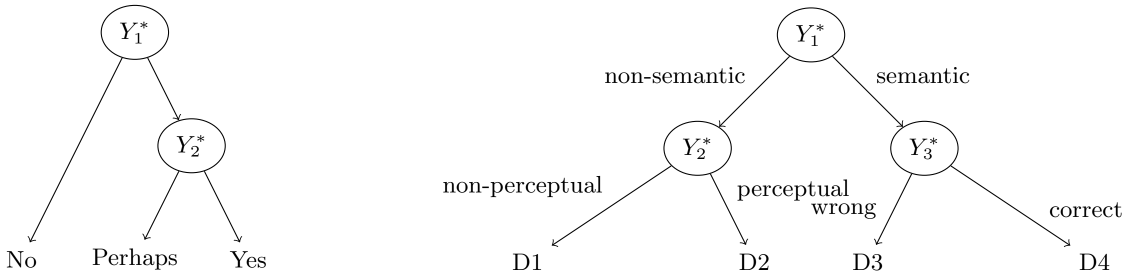

Figure 1: Examples of a linear response tree (left) and a nested response tree (right).

The probabilities of the left and right branch from each node, the two values of the Y^* nodes, can be modeled with a logit link or a probit link. The left and right probabilities are a function of the respondent and the item. For example, for the odds of right vs. left, logit_{pi} (or probit_{pi}) $= \theta_p + \beta_i$, with θ_p as the propensity of person p for the right branch, and β_i as the intensity of the right branch induction by item i. The propensity as well as the induction intensity may depend on the node: $\theta_{p1} + \beta_{i1}$ is used for Y_1^* , $\theta_{p2} + \beta_{i2}$ for Y_2^*, etc. The total model will be presented later, in equations (1) and (2).

Linear response trees are of interest for various kinds of test data and for decision making data.

Linear response category designs

Items of personality and attitude tests often have a point-scale response format with categories that are assumed to be ordered with respect to the latent trait to be measured, such as “strongly agree”, “agree”, “disagree”, and “strongly disagree”, or “no”, “perhaps”, and “yes”, as in Figure 1. Items of ability tests and achievement tests are sometimes multiple choice items with response categories that can be ordered based on their quality (degree of correctness). In all these cases, it is commonly assumed that the categories can be ordered with respect to an underlying latent trait. The higher the category, the more of the latent trait is required to reach the category. Ordered-category data can be modeled with the partial credit model or the graded response model. They rely on the adjacent ratio (e.g., 1 vs. 0, 2 vs. 1, etc.) and the cumulative ratio (e.g., higher than 0 vs. 0, higher than 1 vs. 0 and 1, etc.), respectively. A less common alternative is the continuation ratio model or sequential model (e.g., 1 and higher vs. 0, 2 and higher vs. 1, etc.), see Tutz (1990). Here we follow the latter approach. An overview of all three approaches is given by van der Ark (2001).

The continuation ratio model is a linear response tree model with the same latent variable for all nodes. However, comparing unidimensional and multidimensional linear response tree models, one can investigate whether the response categories actually rely on one underlying trait and thus whether they are ordered with respect to the underlying latent variable indeed. If the same latent person variable does not apply to all nodes, then the categories cannot be seen as ordered although the tree is still linear, and one is then confronted with a measurement problem. It means that depending on where on the response scale a response is given determines which latent trait is being measured. For a similar rationale within the partial credit approach, see Wang and Wu (2011).

Linear response formats

Explicit serial elimination response formats also fulfill the condition of a linear tree. For example, the format of an item such as "Have you ever been in situation A? if yes, have you felt emotion X? yes or no", can be represented with a response tree that is formally equivalent with the left one in Figure 1. Each response step can be based on a different latent trait and a different set of item parameters. For example, how easily one is confronted as a person with a type of situations (top node) is different from how much one is inclined to react with a given emotion (bottom node), and situations can differ in how common they are (top node) as well as in how much they induce a given emotion (bottom node). With a linear response tree model it is possible to separate the occurrence of situations (possibly because of one's own choice) from behavior tendencies in these situations, which is an important issue in the study of personality and emotion (Buss 1987; Emmons *et al.* 1986).

Missing responses

Suppose that the responses to items of an ability test are coded in a binary way, correct vs. incorrect, but that some of the responses are missing. It is common practice to count missing responses as incorrect responses or to consider them as missing at random. Both choices rely on untested assumptions (missing is incorrect, missing is missing at random). An alternative is to consider missing responses as a third response category, so that for the three categories (omission, incorrect, correct) a linear response tree applies. Either a response is omitted or it is not (top node), and when omitted it is correct or incorrect (bottom node). This makes for a linear tree which is formally equivalent to the left one in Figure 1. One advantage of the tree model is that the ability can be measured as a latent trait that is possibly different from an omission tendency and that also the correlation between the two can be estimated. In this way, the tree model is a model for missing not at random. It allows for finding out how much omissions depend on the respondents (e.g., their ability) and on the items (e.g., their difficulty). An application of linear response trees can be found in Holman and Glas (2005). The issue of missing data in a test is even more complicated. The approach just sketched is for intermediate omissions and not for end omissions (items not reached), but also the latter kind can be incorporated in a linear response tree (Glas and Pimentel 2008).

Sequential choice processes

A choice process is sequential if the alternatives are eliminated in a sequential way. For example, suppose that the quality of a product is highly correlated with its price. Then it makes sense that one starts to eliminate from the best and most expensive alternatives up to the level one can afford. The resulting response tree is again of the type in the left part of Figure 1.

Nested response trees are also of interest for various kinds of test data and for decision making data. Linear trees are nested trees as well, but here we will use the label "nested" for nested trees that are not linear.

Hierarchical response category designs

For example, Wright *et al.* (2008) have constructed analogy items with four response categories (A is related to B such as C is related to which of the following? D1, D2, D3, D4). Two of the

D responses are semantically related to the C term. One of both is correct, while the other is not. Two other D responses are not semantically related to C. One of both is visually related to C, while the other is unrelated. The corresponding response tree is a nested tree and is shown in the right part of Figure 1. Instead of an analysis in terms of accuracy only, one can use an IRTree model to measure three different latent variables, one per node in the tree: a semantic propensity (right branch from the top node), a perceptual propensity given a non-semantic approach (right branch from the bottom left node), and accuracy given a semantic approach (right branch from the bottom right node), and also the covariance structure of the three can thus be estimated. Similarly also a different set of item parameters can be defined per node. This type of analogy items has been used in a brain imaging study to investigate in which regions of the brain analogical reasoning leads to higher activity (Wright *et al.* 2008). When the estimates of an IRTree model are correlated with brain activation data as available from the study in question, it is possible to differentiate the regions of specific analogical reasoning from regions that are activated for semantic similarity reasoning more in general and from perceptual similarity reasoning.

Hierarchical response formats

For example, Steinberg and Monahan (2007) have developed a questionnaire to measure autonomy vs. dependence among peers. They use a response format where first a choice has to be made between an autonomous and a dependent response, and second, given one's choice one is asked to indicate how much the choice is endorsed ("sort of true" vs. "really true"). The resulting response tree is formally identical to the one in Figure 1, right, and does not necessarily support the four-point scale that is proposed by the authors for the questionnaire. Perhaps three different latent variables are involved: autonomy vs. dependence, feeling strong about autonomy, and feeling strong about dependence. It is an interesting research issue whether determination is different depending on the kind of initial response and how determination is related to the latent trait of autonomy vs. dependence as measured through the top node. It is also possible to find out how much the item formulation contributes to determination, conditional upon the kind of content (autonomy vs. dependence), separately from how much it induces an autonomy vs. dependence response.

Parallel response formats

For example, Partchev and De Boeck (2012) work with parallel data: accuracy and response times of responses to two intelligence tests. The authors have defined a response tree with a top node for the overall split between fast and slow responses (based on two different median splits for response times), with a left bottom node for the slow responses (slow and correct vs. slow and incorrect), and a right bottom node for the fast responses (fast and correct vs. fast and incorrect). The resulting response tree is again equivalent with the right one in Figure 1. Based on this formulation, Partchev and De Boeck (2012) found that three different latent traits are required to fit the data: speed, slow intelligence, and fast intelligence, although the latter two are rather highly correlated.

Nested choice processes

A choice process is nested if the alternatives are eliminated subset-wise in a sequential way. For example, when deciding on a product, the first choice may be based on the price, while

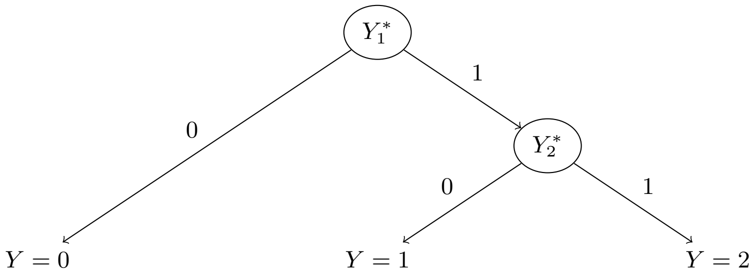

Figure 2: A linear response tree for three response categories.

perhaps for the lower prices a minimum of quality matters, while for the higher prices, given that the quality is guaranteed, perhaps the aesthetic value matters. The resulting response tree is again of the type in the right part of Figure 1.

In general, the IRTree models, based on linear or nested response trees, allow for a different latent variable for each split between the categories (for each internal node of the tree), and also for a different set of item parameters depending on the split (on the node in the tree). First, this makes it possible to measure in a more differentiated way than with a simple correct-incorrect coding or with a classical ordered-category model (partial credit model, graded response model). There are also more assumptions needed for the latter and certainly for the use of a point scale. Second, substantive research questions can be answered related to the meaning of the categories and the underlying response processes, such as whether fast and slow intelligence can be differentiated, and where in the brain analogical reasoning versus semantic similarity reasoning more in general is located.

IRTree models are also useful beyond the classical domain of item response models, as explained for decision making tasks. The processes implied in some cognitive tasks can be also approached with the kind of response trees as discussed. One example are multinomial processing trees (Batchelder and Riefer 1999), on the condition that not two or more branches of the tree connect with the same response category. This is perhaps not the common case, but, when it applies, one can easily include individual differences and (random) task differences with an effect on the response probabilities. Other examples are tasks where also the response time is registered besides the binary response, such as in the Stroop interference tasks (MacLeod 1991) and in the implicit association task (Greenwald *et al.* 1998).

2.2. Response tree model formulation

First, the *linear tree models* will be explained, and next, the nested tree models. Take as an example a test with ten binary items. The responses are either correct or incorrect, but some of the responses are missing (NA), so that there are three response categories (0 = NA, 1 = incorrect, 2 = correct). The corresponding response tree is shown in Figure 2. The left branch from the top node leads directly to NA, while the right branch leads to a further internal node. Let us call the first node sub-item 1, and code the left branch with 0 and the right branch with (1) Going down to the second internal node, the left branch from there on leads to an incorrect response and the right branch leads to a correct response. Let us call the second node sub-item 2, and code the left branch again with 0 and the right branch again with (1)

Each of the three original item responses can now be recoded in terms of the sub-items. NA is recoded into (0, NA) because the first sub-item score is 0 while the second sub-item score is missing. Further, 0 is recoded into (1, 0), and 1 into (1, 1). Suppose the responses of the first person to the ten items are (1, 1, 0, NA, 1, 0, 1, 1, NA, 0). After a recoding in terms of sub-items, this would lead to ((1, 1), (1, 1), (1, 0), (0, NA),(1, 1), (1, 0), (1, 1), (1, 1), (0, NA), (1, 0)).

Let us denote the original item responses as $Y_{pi} = 0, 1, \ldots, M-1$, with $p = 1, \ldots, P$ for persons, $i = 1, \ldots, I$ for items, and $m = 0, \ldots, M-1$ for response categories. The recoded sub-item responses are denoted as $Y^*_{pir} = \mathrm{NA}, 0, 1$, with $r = 1, \ldots, R$ as an index for the sub-items, one per node. The probabilities of the left and right branches from each node, $\pi(Y^*_{pir} = 0, 1)$, are determined by a logistic regression model, or alternatively by a probit model, with a contribution from the person (θ_{pr}), which is a node-specific ability or more generally a latent person variable, and a contribution from the sub-item r nested within the item i (β_{ir}), which is minus the item difficulty or more generally the threshold. Typically, the person contribution is random (a random intercept) and the contribution from the item is fixed (but it can also be random). As a result:

$$\pi(Y^*_{pir} = y^*_{pir}|\boldsymbol{\theta}_p) = \exp(\theta_{pr} + \beta_{ir})^{y^*_{pir}}/[(1 + \exp(\theta_{pr} + \beta_{ir})]. \tag{1}$$

Applied to the whole item i with the three categories from the example, this leads to

$$\pi(Y_{pi} = \mathrm{NA}|\boldsymbol{\theta}_p) = \pi(Y^*_{pi1} = 0|\theta_{p1}), \tag{2a}$$

$$\pi(Y_{pi} = 0|\boldsymbol{\theta}_p) = \pi(Y^*_{pi1} = 1|\theta_{p1})\pi(Y^*_{pi2} = 0|\theta_{p2}), \tag{2b}$$

$$\pi(Y_{pi} = 1|\boldsymbol{\theta}_p) = \pi(Y^*_{pi1} = 1|\theta_{p2})\pi(Y^*_{pi2} = 1|\theta_{p2}), \tag{2c}$$

with $\boldsymbol{\theta}_p = (\theta_{p1}, \theta_{p2})$ and $\boldsymbol{\theta}_p \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_\theta)$. For the given example, $-\theta_{p1}$ must be interpreted as the omission propensity, and $-\beta_{i1}$ as the intensity of an omission induction by the item, while θ_{p2} corresponds to the ability one intends to measure and β_{i2} corresponds to the easiness (minus the difficulty) of a correct response. The correlation between the two latent variables, θ_{p1} and θ_{p2}, is minus the correlation between the ability and the tendency to omit responses. If one would define the contribution of the items also as random, $\boldsymbol{\beta}_i \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$, the resulting correlation informs us about the association of item non-omissions and item easiness. With the items as random effects, one would need to include a fixed effect of the node in (1), δ_r.

The model as defined in (1) and (2a)—(2c) can easily be extended to more than two nodes. Although it is presented here more generally than for ordered categories, it can of course also be used also for that purpose, and thus for multiple choice items where commonly the partial credit model (Masters 1982) or the graded response model (Samejima 1969) are used for. However, the linear response tree model is not equivalent with these models, neither in the formal sense because it relies on differently defined odds (continuation ratio), nor in the process sense, because the tree model assumes a covert or overt sequential process. This process is different, for example, from a divide-by-total process, which for choices is also called the Bradley-Terry-Luce choice rule (Bradley and Terry 1952; Luce 1959). The divide-by-total rule is the underlying rationale for the partial credit model. Although the latter can also be represented with a tree, as shown by Revuelta (2008), the tree is a feature tree, one that does not lead to a GLMM. The measurement qualities of the unidimensional linear response tree model are discussed by Hemker *et al.* (2001).

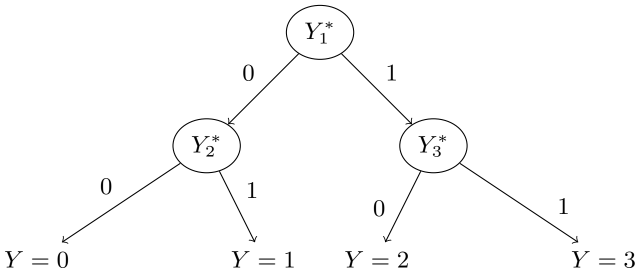

Figure 3: A nested response tree for four response categories.

For a *nested tree model*, the formulation is analogous. The equation per node is the same as in (1). Suppose the tree is as presented in Figure 3, with four response categories, from 0 to 3, going from left to right, then the following equations apply for the four response categories:

$$\pi(Y_{pi} = 0|\boldsymbol{\theta}_p) = \pi(Y^*_{pi1} = 0|\theta_{p1})\pi(Y^*_{pi2} = 0|\theta_{p2}), \tag{3a}$$

$$\pi(Y_{pi} = 1|\boldsymbol{\theta}_p) = \pi(Y^*_{pi1} = 0|\theta_{p1})\pi(Y^*_{pi2} = 1|\theta_{p2}), \tag{3b}$$

$$\pi(Y_{pi} = 2|\boldsymbol{\theta}_p) = \pi(Y^*_{pi1} = 1|\theta_{p1})\pi(Y^*_{pi3} = 0|\theta_{p3}), \tag{3c}$$

$$\pi(Y_{pi} = 3|\boldsymbol{\theta}_p) = \pi(Y^*_{pi1} = 1|\theta_{p1})\pi(Y^*_{pi3} = 1|\theta_{p3}), \tag{3d}$$

with $\boldsymbol{\theta}_p = (\theta_{p1}, \theta_{p2}, \theta_{p3})$ and $\boldsymbol{\theta}_p \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_\theta)$. Again, the left branches from each node are coded with 0 and the right branches are coded with 1. For the slow and fast intelligence example, Y^*_{pi1} refers to the top node, with $Y^*_{pi1} = 0$ for a slow response, while $Y^*_{pi1} = 1$ for a fast response. The probability of left and right is determined by the respondent's speed (θ_{p1}) and by the speediness of the item (β_{i1}). Y^*_{pi2} refers to the bottom left node, and the probability of left (slow and incorrect) and right (slow and correct) is determined by the respondent's slow intelligence (θ_{p2}) and how easy the item is when the response is slow (β_{i2}). Y^*_{pi3} refers to the bottom right node, and the probability of left (fast and incorrect) and right (fast and correct) is determined by the respondent's fast intelligence (θ_{p3}) and how easy the item is when the response is fast (β_{i3}). If one would define the contribution of the items also as random, $\boldsymbol{\beta}_i \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$, the resulting correlation informs us about the association of item speed, slow easiness and fast easiness. With the items as random effects, one would need to include a fixed effect of the node in equation (1), δ_r.

The more general equation can be formulated with the help of a $(M-1) \times R$ mapping matrix $\mathbf{T}$ that indicates for each end node (response category) with which internal nodes it is connected: $T_{mr} = 0$ for a connection through the left branch, and $T_{mr} = 1$ for a connection through the right branch, respectively, and T_{mr} is NA when there is no connection. The $\mathbf{T}$ matrices for the examples of Figures 1 (left part) and 2, and for Figures 1 (right part) and 3 are given in Table 1, left and right, respectively. The resulting formula is

$$\pi(Y_{pi} = m|\boldsymbol{\theta}_p) = \prod_{r=1}^{R} \left[\frac{\exp(\theta_{pr} + \beta_{ir})^{t_{mr}}}{1 + \exp(\theta_{pr} + \beta_{ir})} \right]^{d_{mr}}, \tag{4}$$

with $t_{mr} = T_{mr}$, except that $t_{mr} = 0$ if T_{mr} is missing; with $d_{mr} = 0$ when T_{mr} is missing and $d_{mr} = 1$ otherwise.

The linear response tree model is formally equivalent with discrete survival models (Santos and Berridge 2000; Singer and Willett 1993; Willett and Singer 1995).

$$\text{(linear)} \quad \begin{array}{c|cc|} & Y_1^* & Y_2^* \\ \hline Y{=}0 & 0 & - \\ Y{=}1 & 1 & 0 \\ Y{=}2 & 1 & 1 \end{array} \qquad \text{(nested)} \quad \begin{array}{c|ccc|} & Y_1^* & Y_2^* & Y_3^* \\ \hline Y{=}0 & 0 & 0 & - \\ Y{=}1 & 0 & 1 & - \\ Y{=}2 & 1 & - & 0 \\ Y{=}3 & 1 & - & 1 \end{array}$$

Table 1: Mapping matrices **T** for the linear and the nested tree.

2.3. Data preparation and data analysis with response trees

We provide example data sets illustrating some of the models. For convenience, these are part of the package called **irtrees**. The package also includes two functions, `dendrify` and `exogenize`, to prepare the data for analysis. Function `dendrify`, which is useful with response tree models, will be described briefly here, and `exogenize` will be discussed in Section 3 on latent-variable tree models.

Item response data typically require some preparation before they can be submitted to `glmer`. This is true even in the simpler case when all responses are dichotomous (De Boeck *et al.* 2011). Many researchers will have their data in so-called wide form, where each individual represents a row in a data matrix, and each item occupies a column. However, `glmer` expects the data in long form, where each row of the data matrix pertains to the encounter of person p with item i. In the simplest case when there are no person or item covariates, each line in a long-form data set needs to contain only three columns: person ID, item ID, and response. As a general tool to convert between long and wide form, we recommend package **reshape** (Wickham 2007).

With the models discussed in this article, there is a further complication as each response must be broken down into sub-responses, such that each line of the data matrix will now pertain to a person and a sub-item (an item node). When all the items have the same number of categories and hence the same mapping matrix, this is relatively easy: (1) reshape the original responses in wide form to long form (one response per row), (2) apply the mapping matrix to the response, resulting in a new wide form with as many new columns as there are nodes, and (3) reshape again into long form, now with one node per line. This is what function `dendrify` does. It accepts two arguments:

- a data matrix, which is assumed in wide form, and with items all of the same type, i.e., the same mapping matrix applies to all items. For efficiency reasons, the data is required to be be a matrix of integers, not a data frame. A data frame can be converted to a data matrix with function `data.matrix()`, which replaces any factors and ordered factors with their integer representations; and
- a matrix containing the mapping matrix to be applied to the original responses.

The output is a data frame that can be analyzed with `glmer`; see the **irtrees** manual for further detail.

Data

The simulated data for the *linear response tree* are accuracy data for an example of an ability test with missing responses, and with $N = 300$ and 10 items. The response categories after the inclusion of missing data are: NA (missing), 0 (incorrect), and 1 (correct). The data matrix is called `linresp`. The data are generated with:

- a bivariate normal distribution for the latent person variables: zero means, variance of θ_{p1} and θ_{p2} equal to 0.50 and 1.00, respectively, and a correlation of 0.50, such that the more able respondents omit less responses;
- a bivariate normal distribution for the items: zero means, variance of β_{i1} and β_{i2} equal to 1.00, and a correlation of 0.70, such that the omission rate decreases with the easiness of the items;
- fixed node effects of 1.00 for the response vs. omission node (responses are more likely than omissions) and 0.00 for the correct vs. incorrect node (the chances are on average equal for correct and incorrect responses).

Because of the random generation process, the best fitting values for the data are commonly different from the generating values, also when the true model is estimated. This is also true for the other simulated data examples.

The simulated data for the *nested response tree* are data with four categories: 0, 1, 2, and 3. The nested structure is presented in Figure 3. The size of the data matrix is again $N = 300$ respondents by 10 items. The data matrix is called `nesresp`. The data are generated with:

- a trivariate normal distribution for the latent person variables: zero means, variance of θ_{p1}, θ_{p2}, and θ_{p3} equal to 1.00, and a uniform correlation of 0.50, such that the respondents with a higher speed have a higher slow intelligence and also a higher fast intelligence;
- a trivariate normal distribution for the items: zero means, variance of β_{i1}, β_{i2}, and β_{i3} equal to 1.00, and again a uniform correlation of 0.50, such that the faster items are easier to solve when working slow and when working fast;
- fixed node effects of 0.00 for the top node (fast vs. slow response), 0.50 for slow accuracy (slow is easier) and -0.50 for fast accuracy (fast is more difficult).

Preparing the data for analysis

The example data sets have been supplied as integer data matrices (not data frames) in wide form. To transform them into the form required by `glmer`, we must first define the mapping matrices which map the item responses on sub-item responses. Each mapping matrix has as many rows as there are response categories and as many columns as there are sub-items (nodes). The mapping matrix for the linear tree and nested tree examples are given in Table 1. In R:

```
R> linmap <- cbind(c(0, 1, 1), c(NA, 0, 1))
R> nesmap <- cbind(c(0, 0, 1, 1), c(0, 1, NA, NA), c(NA, NA, 0, 1))
```

Following that, all we need to do is call the `dendrify` function:

```
R> linrespT <- dendrify(linresp, linmap)
R> nesrespT <- dendrify(nesresp, nesmap)
```

The resulting data frame contains the binary sub-item responses (`value`), a factor for the original items (`item`), a factor for the respondents (`person`), a factor for the nodes (`node`), and a factor for the sub-items, as many as there are items times nodes (`sub`).

The `glmer` *code*

The `glmer` code will not be explained in detail here. The function is interchangeable with the `lmer` function and the latter is explained in detail by De Boeck *et al.* (2011) and Doran *et al.* (2007). If a non-default option is chosen for `lmer` (e.g., `family = binomial` for binary data), then the function `glmer` is called, and if the default option is chosen with `glmer` (the `family` argument is omitted or `family = gaussian`), then the `lmer` function is called. Note that choosing the restricted maximum likelihood (`REML`) option has an effect only when combined with the default option (`gaussian`), and that adaptive Gaussian quadrature (`nAGQ = 2` or larger) can be chosen instead of the Laplace approximation (`nAGQ = 1` or the `nAGQ` option is omitted). For a benchmark data set (Tuerlinckx *et al.* 2004) included in the **lme4** package (`VerbAgg`), the adaptive quadrature as implemented for `glmer` does not seem to reduce the shrinkage of the variance estimates compared to other Gaussian adaptive or non-adaptive quadrature implementations. Shrinkage is typical for approximative methods such as the Laplace approximation (Joe 2008). It is unclear why the adaptive quadrature does not have the desired effect of shrinkage reduction. Because the shrinkage is after all rather minor from about 10 items on, and because the Laplace approximation is faster than adaptive quadrature, we will omit the `nAGQ` argument.

In the following, the `glmer` code is given first for four different *linear response tree* models:

(1) The multidimensional model for linear response trees

```
R> M1linTree <- glmer(value ~ 0 + item:node + (0 + node | person),
+    family = binomial, data = linrespT)
```

The term `item:node` tells the function that the individual fixed sub-item effects (10 × 2) must be estimated and thus a set of item parameters (10) per node (2). The term `(0 + node | person)` causes the variances and covariances of the two latent variables (non-omission tendency and ability) to be estimated. If one is interested in the empirical Bayes estimates, then the `ranef` function can be used to produce the posterior mode estimates, and this holds also for all following models:

```
R> ranef(M1linTree)
```

Suppose that the three categories are “no”, “perhaps”, and “yes”, then the two latent variables refer to the two choice points on the scale: not disagreeing (“yes” or “perhaps” vs. “no”) and an affirmation (“yes” vs. “perhaps”). As explained before, such a differentiation can make sense and can be interpreted depending on the content of the items. In the case of categories referring to increasing degrees of agreement or increasing quality of the response, it is important to test the unidimensional model, with one underlying latent variable instead of a different one per node or step on the response scale. An illustration of this approach is given in the real data application, and, for the `linrespT` data the code is given hereafter.

(2) The unidimensional model for linear response trees

```
R> M2linTree <- glmer(value ~ 0 + item:node + (1 | person),
+    family = binomial, data = linrespT)
```

The difference with the previous is that now both nodes have the same latent variable.

One further simplification would be that the distance between the points on the scale is the same for all items. This is equivalent with a main effect of node without an interaction of node with item. It is the rating scale version (Andrich 1978) of the model for linear response trees.

(3) The unidimensional rating scale model for linear response trees

```
R> M3linTree <- glmer(value ~ 0 + item + node + (1 | person),
+    family = binomial, data = linrespT)
```

This reduces the number of fixed effect parameters from 20 to $10 + (2 - 1)$.

Suppose that the items are modeled with random effects, then a multivariate distribution can be estimated also for the items. This can be helpful for the missing responses application, because it allows us to estimate the correlation between item easiness and item non-omission as a parameter of the model, and also to introduce item covariates to explain the difficulties and the omissions while still allowing for a residual in the explanatory model. The code for random items and a multidimensional model is as follows.

(4) The multidimensional random item model for linear response trees

```
R> M4linTree <- glmer(value ~ 1 + (0 + node | item) + (0 + node | person),
+    family = binomial, data = linrespT)
```

The term `(0 + node | items)` leads to the estimation of an item variance per node and an item correlation matrix for the nodes. If one would be interested in the effect of a covariate (`cov`), a term `cov * node` can be included.

The code for two different *nested response tree* models will be described next:

(5) The nested tree model with a dimension per node

```
R> M5nesTree <- glmer(value ~ 0 + item:node + (0 + node | person),
+    family = binomial, data = nesrespT)
```

As a result, a three-dimensional distribution will be estimated for the persons as well as three item parameter sets, one for each node.

(6) The nested tree model with lower dimensionality

Suppose one wants to estimate a model with only one intelligence, the same for slow and fast responses, then one should first define an alternative node factor where fast and slow are collapsed:

```
R> nesrespT$node2 <- factor(nesrespT$node != "node1")
```

Suppose further that the item easiness for slow and fast responses differ only with respect to an additive constant. That would require an additional recoding:

```
R> nesrespT$fs <- as.numeric(nesrespT$node == "node3")
```

The code for such a model is as follows:

```
R> M6nesTree <- glmer(value ~ 0 + item:node2 + fs + (0 + node2 | person),
+    family = binomial, data = nesrespT)
```

The term `(0 + node2 | person)` will produce estimates for the variances of speed and of the common underlying ability for fast and slow accuracy, as well as for their correlation. Because of the term `item:node2` the item parameters for speed and accuracy are estimated, and finally, because of the term `fs` also the fixed effect of fast vs. slow is estimated. The latter effect is negative, as may be expected on the basis of the data generation.

2.4. Response tree applications with real data

Are the categories ordinal?

The first application concerns *linear response tree* modeling. The data are responses of 316 subjects to a questionnaire on verbal aggression with 24 items and three response categories per item: "no", "perhaps", and "yes" (De Boeck and Wilson 2004). The data set is included in the **lme4** package and is labeled `VerbAgg`; we have included it in wide-form in **irtrees**. Various models for ordinal data have been fit to these data (De Boeck and Wilson 2004). However, for the ordering to make sense, a "yes" response needs to be an indication of a stronger verbal aggression tendency than a "perhaps" response, and similarly for a "perhaps" response compared to a "no" response. This is the ordinal hypothesis and it has not been tested thus far for these data. It corresponds with a linear response tree with two nodes and the same latent trait for both nodes, for the differentiation between "no" vs. "perhaps or yes" and for the differentiation between "perhaps" vs. "no", while the item parameters may differ depending on the node. The ordinal hypothesis can be tested by comparing the goodness of fit of the one-dimensional model (a common latent trait for both nodes) with the two-dimensional model (a different latent trait per node).

The choice for "perhaps" or "yes" vs. "no" which refers to the top node is about admitting one may show an aggressive reaction. Admission is different from affirmation. The choice for "yes" vs. "perhaps" refers to the second node and implies the affirmation of an aggressive reaction. Admission and affirmation may be two qualitatively different responses and rely on different latent traits. Affirmation is not necessarily more of the same or just stronger than admission. The response tree is formally equivalent with the one in Figure 2.

Package **irtrees** includes a data matrix `VerbAgg3` with the trichotomous responses (no, perhaps, yes). It can be dendrified for the IRTree analysis with a linear response tree as follows:

```
R> mapping <- cbind(c(0, 1, 1), c(NA, 0, 1))
R> VerbAgg3T <- dendrify(VerbAgg3[ , -(1:2)], mapping)
```

The `glmer` code for the two models is as follows. One-dimensional:

```
R> onedim <- glmer(value ~ 0 + item:node  + (1 | person),
+    family = binomial, data = VerbAgg3T)
```

Two-dimensional:

```
R> twodim <- glmer(value ~ 0 + item:node + (0 + node | person),
+    family = binomial, data = VerbAgg3T)
```

As can be derived from the code, items and nodes are included in the model with fixed effects.

Based on the AIC (Akaike, 1974) and BIC (Schwartz, 1978), the two-dimensional model fits clearly better (AIC = 12378, BIC = 12751) than the one-dimensional model (AIC = 12778, BIC = 13137). The correlation between the two latent traits of the former model is only 0.392. This is rather strong evidence against a quantitative ordering interpretation of the three response categories and in favor of a qualitative difference between the latent traits involved in the same response scale. Whether this result can be generalized to other seemingly ordered responses is not clear, but it deserves further investigation.

Can fast and slow intelligence be differentiated?

The second application concerns *nested response tree* modeling and is reported in Partchev and De Boeck (2012). The data are binary accuracy data (correct, incorrect) and response times of responses to two types of intelligence test items: Raven-like matrices (35 items, $N = 503$) and verbal analogies (36 items, $N = 726$). All respondents are men aged 17 to 27. The research issue is whether the intelligence underlying slow responses is the same as the intelligence underlying fast responses. The answer to this question is of importance from a measurement point of view as well as for the study of cognitive processes of intelligence.

A nested tree is defined for the combination of response time and accuracy. The first division is between fast and slow responses, based on a median split of response times. For reasons of robustness, the median split is performed within persons as well as within items, and the results of the analysis are very similar for the two splits. The second division is within the slow responses between correct and incorrect responses, while the third division is analogous to the second but now within the fast responses. The tree is formally equivalent with the one in Figure 3. The top node is for speed of responding, the two nested nodes are for slow and fast accuracy, respectively.

The differentiation hypothesis (fast vs. slow) can be tested with respect to the persons as well as with respect to the items. For the persons, differentiation implies that two different abilities are involved, whereas lack of differentiation means that the same ability is involved but perhaps more (or less) of it in fast accuracy than in slow accuracy. For the items, differentiation means that two qualitatively different difficulty profiles are involved, while lack of differentiation means that the profiles differ at most with respect to their level. Qualitatively different profiles point to differences in the underlying processes. In total, four models can be formulated: (1) fast and slow are differentiated with respect to ability and difficulties, (2) only with respect to difficulties, (3) only with respect to abilities, (4) neither with respect to abilities, nor difficulties, which implies that fast and slow intelligence are the same.

The mapping matrix to be used for the four models is the following:

```
R> mapping <- cbind(c(0, 0, 1, 1), c(0, 1, NA, NA), c(NA, NA, 0, 1))
```

The factor `node` that results from the `dendrify` function is recoded into a factor `node2` for the case there is no differentiation between fast and slow for the persons or for the items, or for neither of both, such that `node2` has only two levels (one for speed, and another for accuracy independent of whether the response is fast or slow). Suppose that the data frame is called `fsdatT` (not part of the **irtrees** package), then

```
R> fsdatT$node2 <- factor(fsdatT$node != "node1")
```

For the case there is no differentiation between fast and slow with respect to items, there may still be a main effect, for example because fast is more difficult overall. Therefore, it helps to define the following factor:

```
R> fsdatT$fs <- as.numeric(fsdatT$node == "node3")
```

The code for the four models is as follows. Double differentiation:

```
R> doublediff <- glmer(value ~ 0 + item:node + (0 + node | person),
+    family = binomial, data = fsdatT)
```

Difficulty differentiation:

```
R> diffdiff <- glmer(value ~ 0 + item:node + (0 + node2 | person),
+    family = binomial, data = fsdatT)
```

Ability differentiation:

```
R> abildiff <- glmer(value ~ 0 + item:node2 + fs + (0 + node | person),
+    family = binomial, data = fsdatT)
```

No differentiation:

```
R> nodiff <- glmer(value ~ 0 + item:node2 + fs + (0 + node2 | person),
+    family = binomial, data = fsdatT)
```

Following Partchev and De Boeck (2012) the double differentiation model turned out to be the best fitting model for matrices and for verbal analogies, although the correlation between the fast and slow abilities is close to 0.90. The conclusion is that fast and slow intelligence are different but not much different. This kind of paradigm and modeling can be applied also to other intelligence tests and other kinds of tasks.

3. Latent-variable tree models

3.1. Why latent-variable tree models?

The structural part of a structural equation model can be formulated as a network of latent variables. The network tells us which latent variables can be written as a function of which other latent variables. In order for the structural part of a model to be formulated as part of a GLMM it is sufficient that it fulfills the criteria of a latent-variable tree of the kind to be specified hereafter. The *end-node* latent variables are defined as the (unweighted) sum of all *internal-node* latent variables they are connected with in the upward direction in the tree. If such a tree formulation is possible (not necessarily a binary tree), then again general-purpose software for GLMMs can be used.

The latent-variable trees can be linear or nested. An example of a *linear tree* is given in the left part of Figure 4. The end nodes are denoted as θ_r, with $r = 1, \ldots, R$ as an index for the end nodes, while the nodes higher up are internal nodes and denoted as θ'_s, with $s = 1, \ldots, S$.

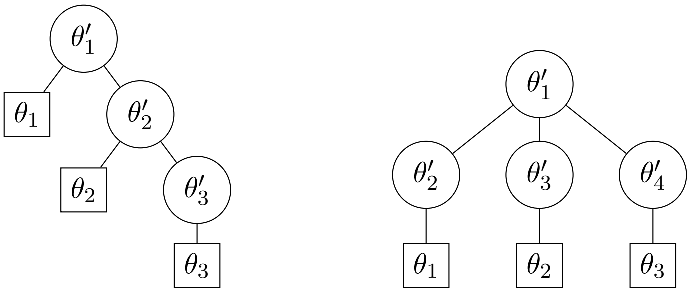

Figure 4: A linear latent-variable tree (left) and a nested latent-variable tree (right).

The branches of a latent-variable tree are not represented with arrows in order to indicate that their meaning is different from the meaning of the branches in a response tree. The idea behind the left tree in Figure 4 is a cumulative process such that the each of the end nodes is the sum of all internal nodes it is connected with. As a consequence, it follows from Figure 4 that $\theta_3 = \theta'_1 + \theta'_2 + \theta'_3$, and also that $\theta'_3 = \theta_3 - \theta_2$. In other words, the internal-node latent variables are difference latent variables. When the internal-node latent variables are allowed to correlate, this structure corresponds to the structural part of Embretson's (1991) model for learning and change. The cumulative process is the latent variable equivalent of what is called an integrative process in time series models. It will be shown in the section on model formulation that also other types of serial dependency between end-node latent variables can be modeled with a latent-variable tree (i.e., without the assumption of an integrative process). Serial dependency between latent variables is a recent topic of investigation in growth curve latent variable modeling (e.g., Hung 2010).

An example of a *nested tree* is given in the right part of Figure 4. Nested trees for latent variables are of interest, for example, for the structure of intelligence. For example, in a well-known article, Gustafsson (1984) describes a model for intelligence with three levels: (1) general intelligence (g) on top (level 3), (2) broad intelligence dimensions such as Fluid and Crystallized Intelligence in the middle (level 2) with loadings on g, and (3) factors called Group factors at the bottom (level 1) with loadings on the broad intelligence dimensions. The manifest variables load on one or more latent variables from the lowest level. For example, analogy tests load on an Induction factor, Induction loads on Fluid Intelligence and Fluid Intelligence loads on g. A structure with two latent levels is more common, it is called a second-order structure. We will focus here on models with two levels, but an extension to more levels is straightforward.

An alternative for a second-order structure is the bi-factor model. An example of the corresponding tree is given in the right panel of Figure 4. For a description of the bi-factor model, see, for example Cai (2010) and Rijmen (2010). The difference between the regular bi-factor model and the models described here is that here all weights are equal to zero or one. The weight of an end-node latent variable is one for all items associated with the end node, and zero otherwise. The end-node latent variables each receive an unweighted input from (i.e., are the sum of) two internal-node latent variables: first, a general latent variable (top internal node), and second, a latent variable that connects with only one end node and is thus end-node specific (bottom internal nodes). For example, following the right panel of Figure 4, $\theta_1 = \theta'_1 + \theta'_2$. All internal-node latent variables are independent. For an unweighted input model as discussed here, the second-order model is equivalent with the bi-factor model.

The bi-factor model makes sense, for example, if one wants to have an overall measure for various positively correlating abilities. The correlating abilities are the end nodes $(\theta_1, \theta_2, \theta_3)$ and their correlation is induced by the general ability (θ_1) which they all share. The more specific internal-node latent variables $(\theta_2, \theta_3, \theta_4)$ are the differences with the general internal-node latent variable. The advantage of the bi-factor model is that it can be used to study the structure of individual differences in a given domain and that it is an interesting measurement tool, which provides an overall score and also specific scores which can be interpreted in a simple way, for example, as subscale-specific residuals.

3.2. Latent-variable tree model formulation

We will first formulate the model for a *linear latent-variable tree.* The links between the end-node latent variables, θ_r, and the internal-node latent variables, θ_s, are indicated in a $R \times S$ matrix $\mathbf{T}$, such that $T_{rs} = 0$ means that there is no link between θ_r and θ'_s and $T_{rs} = 1$ means that there is a link indeed. The general equation for latent-variable tree models is as follows:

$$\pi(Y_{pir} = 1|\boldsymbol{\theta}'_p) = \exp\left(\sum_{s=1}^{S} \theta'_{ps} T_{rs} + \beta_{ir}\right) / \left[1 + \exp\left(\sum_{s=1}^{S} \theta'_{ps} T_{rs} + \beta_{ir}\right)\right], \tag{5}$$

where Y_{pri} is the response of person p to item i that measures end-node latent variable r;

where T_{rs} is the value of element (r, s) in $\mathbf{T}$;

where θ'_{ps} denotes the latent variable of internal node s, $\boldsymbol{\theta}'_p \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\theta'})$;

where β_{ir} is minus the item difficulty of item i for end node r.

The notation is for the case all items are crossed with the index r, for example, when all items are presented at each point in time. However, it is also possible that the items are nested in the values of r, and for that case the notation $r[i]$ is used: $Y_{pr[i]}$ and $\beta_{r[i]}$. The β_{ir} can account for a shift in the mean depending on r. Alternatively, one may use a parameterization $\beta'_{ir} + \mu_r = \beta_{ir}$, where μ_r is the end-node r specific mean and β'_{ir} is the item-specific deviation.

In time-series models a distinction is made between integrative processes, autoregressive processes, and moving average processes. *Integrative processes* correspond to a linear tree as explained earlier and as represented in the left part of Figure 4. The corresponding mapping matrix for $R = 3$ is given hereafter:

$$\begin{array}{c c} \text{(linear latent-variable tree)} & \begin{array}{ccc} \theta'_1 & \theta'_2 & \theta'_3 \end{array} \\ \begin{array}{c} \theta_1 \\ \theta_2 \\ \theta_3 \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

The autoregressive (AR) and moving average (MA) processes cannot be represented with such a linear tree. However, for the MA process a reformulation is possible that does lead to a latent-variable tree and a GLMM. In the MA process, an observed value is a weighted sum of previous random inputs and a new random input. For a MA(1) process (lag = 1), only the just preceding random input counts: $\theta_r = \alpha\theta''_{r-1} + \theta''_r$, with θ''_r as the random input at time r. The θ'' notation for the MA process is because the random input variables will be replaced with internal-node latent variables, denoted with θ' later. We will consider here only lag = 1 processes, and the α weights may depend on r, which means that the process is not necessarily a stationary process. The resulting model is called a *free MA(1)* process model. A possible interpretation of the MA(1) process in a learning context is that an individual increase (or decrease) at time r has an effect on the learning outcome at time $r+1$; for example, making progress stimulates for the next trial.

For the free MA(1) process and $R = 3$ as in the example, the formulas are:

$$\theta_1 = \theta''_1, \tag{6a}$$
$$\theta_2 = \theta''_2 + \alpha_{12}\theta''_1, \tag{6b}$$
$$\theta_3 = \theta''_3 + \alpha_{23}\theta''_2. \tag{6c}$$

The α weights seem to make the model into a bilinear model so that it would not seem to fulfill the GLMM conditions. However, the model can be reformulated as a nested latent-variable tree model without weights, and such that the end-node latent variables can each be written as a sum of internal-node latent variables:

$$\theta_1 = \theta'_1 + \theta'_2, \tag{7a}$$
$$\theta_2 = \theta'_2 + \theta'_3 + \theta'_4, \tag{7b}$$
$$\theta_3 = \theta'_4 + \theta'_5. \tag{7c}$$

The reformulation is achieved as follows:

1. Define a $S \times S$ diagonal variance-covariance matrix $\mathbf{\Sigma}_{\theta'}$ for the internal nodes, with $\sigma^2_{\theta'_s}$ on the diagonal of row s, and $S = 2R - 1$.

2. Define a mapping matrix $\mathbf{T}$ such that the consecutive end nodes are mapped onto a moving window of three consecutive internal nodes with an overlap of one, except that the moving window begins and ends with a width of two instead of three (a mapping onto two instead of three internal nodes). For $R = 3$, this leads to the following 3×5 mapping matrix:

$$\begin{array}{c c} \text{(free MA(1))} & \begin{array}{ccccc} \theta'_1 & \theta'_2 & \theta'_3 & \theta'_4 & \theta'_5 \end{array} \\ \begin{array}{c} \theta_1 \\ \theta_2 \\ \theta_3 \end{array} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

   The corresponding tree is shown in Figure 5, and it is not a linear tree (and neither is it a rooted tree).

3. It follows from the previous two points that:

$$\mathbf{\Sigma}_\theta = \mathbf{T}\mathbf{\Sigma}_{\theta'}\mathbf{T}^T. \tag{8}$$

4. From equations (6) and (7), it follows that $\alpha_{r,r+1}\theta''_r = \theta'_{2r}$, $\theta''_r = \theta'_{s=2r-1} + \theta'_{s=2r}$, $\alpha^2_{r,r+1} = \sigma^2_{\theta'_{s=2r}}/(\sigma^2_{\theta'_{s=2r-1}} + \sigma^2_{\theta'_{s=2r}})$. For example, $\alpha^2_{23} = \sigma^2_{\theta'_4}/(\sigma^2_{\theta'_3} + {}^2_{\theta'_4})$.

If the latent variables are time-ordered, a *growth curve model* component does make sense. A simple growth curve model would include an intercept latent variable and a slope latent variable, which is the random effect of a linear time covariate (X_r): $\theta_r = \mu_0 + \theta_0 + X_r(\lambda + \theta_{\text{lin}})$, with μ_0 as the fixed part (mean) of the intercept, θ_0 as the random part of the intercept, λ as

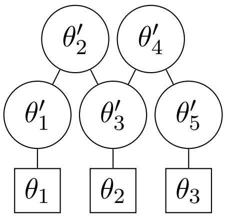

Figure 5: The free MA(1) tree for three moments in time.

the fixed part (mean) of the slope, and θ_{lin} as the random part of the slope or linear growth latent variable. For equally spaced time points, $X_r = r - 1$. The full model has also an independent residual, θ'_r, per time point: $\theta_r = \mu_0 + \theta_0 + (r-1)(\lambda + \theta_{\text{lin}}) + \theta'_r$ (McArdle *et al.* 2009). This residual can be used to model serial dependencies, with an AR process (Hung 2010) or with a MA process.

With a free MA(1) process for the residual latent variable, the linear growth model is as follows:

$$\pi(Y_{pir} = 1) = \frac{\exp\left[\mu_0 + \theta_{p0} + X_r(\lambda + \theta_{p,\text{lin}}) + \sum_{s=1}^{S} T_{rs}\theta'_{ps} + \beta'_{ir}\right]}{1 + \exp\left[\mu_0 + \theta_{p0} + X_r(\lambda + \theta_{p,\text{lin}}) + \sum_{s=1}^{S} T_{rs}\theta'_{ps} + \beta'_{ir}\right]}, \tag{9}$$

where μ_0 as the fixed part of the intercept;

where λ as the fixed part of the slope, the fixed effect of X_r;

where $\boldsymbol{\theta}_p \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_\theta)$, with $\boldsymbol{\theta}_p$ as a symbol for the θs with and without a prime, while the covariance of each θ'_s is zero;

where T_{rs} as the element from the free MA(1) mapping matrix corresponding to end node (time point) r and internal node s;

where β'_{ir} as the r-specific end-node deviation of item i.

For a *nested latent-variable tree* (e.g., the right part of Figure 4) the formulation is identical to the one in equation 5, but with uncorrelated latent variables ($\boldsymbol{\Sigma}'_\theta$ is diagonal). The only difference is the mapping matrix. The mapping matrix corresponding to the right part of Figure 4 is as follows

$$\begin{array}{cc} \text{(bi-factor)} & \begin{array}{cccc} \theta'_1 & \theta'_2 & \theta'_3 & \theta'_4 \end{array} \\ \begin{array}{c} \theta_1 \\ \theta_2 \\ \theta_3 \end{array} & \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \end{array}$$

3.3. Data preparation and data analysis with latent-variable trees

Package **irtrees** includes two example sets for latent-variable trees, one for the linear case and the MA(1) model and one for the nested case and the bi-factor model. They can be prepared for analysis with `glmer` with the help of function `exogenize`.

Similar to `dendrify`, function `exogenize` takes the data (an integer data matrix in wide form) as the first argument, and the mapping matrix as the second argument. There are three further arguments:

`items` is an integer vector indicating which columns of the data matrix (items) to include (default is use all). These columns refer to responses Y_{pir} when the items are crossed with the end nodes (e.g., with end nodes referring to time points), and to responses $Y_{pr[i]}$ when the items are nested in the end nodes (e.g., with end nodes referring to subscales).

`endnode` is a factor with the same length as `items` indicating the end nodes to which the Y_{pir} or $Y_{pr[i]}$ are attached, one end node per Y_{pir} or $Y_{pr[i]}$.

`crossitem` is a factor with the same length as `items` indicating the items i. There are two cases. The first is where i is repeated across end nodes (crossed design). The second is where i is nested in the end nodes (nested design). In the second case, this third argument can be omitted.

Data

The data for the *linear latent-variable tree* are binary data with 300 respondents and 10 items, presented at three moments in time. The relationship between the end-node latent variables per moment in time is defined as a stationary MA(1) process, as follows: $\theta_r = \alpha_{r-1,r}\theta''_{r-1} + \theta''_r + \mu_r$, with $r = 1, 2, 3$; where $\boldsymbol{\theta}'' \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\theta''})$, with $\boldsymbol{\Sigma}_{\theta''}$ as a 4×4 ($r = 0, \ldots, 3$) diagonal matrix with 0.75 on the diagonal, where $\alpha_{r-1,r} = \sqrt{1/3}$, and where $\boldsymbol{\mu}_\theta = (-0.25, 0, 0.25)$. As a result, the expected variance of the three end-node latent variables is 1.00, the expected correlation of adjacent end-node latent variables is 0.50, while the expected correlation of non-adjacent ones is 0.00. The difficulties do not change over time ($\beta_{i1} = \beta_{i2} = \beta_{i3}$), and they are drawn from the standard normal distribution, $\beta_i \sim \text{N}(0, 1)$. The resulting data matrix is called `linlat`.

The data set for the *nested latent-variable tree* is again an array of 300 respondents by 30 items. The items each belong to a subscale of ten items related to the same end node. There are four internal nodes: a general one, and three subscale-specific ones. The data are generated with an independent normal distribution and a zero mean for all θ's, with a variance of 1.00 for the general internal-node latent variable, θ'_1, and variances of 0.50, 1.00, and 1.50 for the three subscale specific internal-node latent variables, θ'_2, θ'_3, θ'_4, respectively. And $\beta_i \sim \text{N}(0, 1)$. The resulting data matrix is called `neslat`.

Preparing the data for analysis

For the *free MA(1)* example, the mapping matrix is

```
R> lmm <- cbind(c(1, 0, 0), c(1, 1, 0), c(0, 1, 0), c(0, 1, 1), c(0, 0, 1))
```

The file is organized such that responses to the ten items at time point 1 are given first, followed by the ten responses at time point 2, etc. We specify this arrangement with the `endnote` and `crossitem` options, and the call to the `exogenize` function becomes

```
R> linlatT <- exogenize(linlat, lmm,
+     endnode = rep(1:3, each = 10), crossitem = rep(1:10, 3))
```

For the *bi-factor* example, the mapping matrix is

```
R> nmm <- cbind(c(1, 1, 1), c(1, 0, 0), c(0, 1, 0), c(0, 0, 1))
```

and the call to exogenize is

```
R> neslatT <- exogenize(neslat, nmm, endnode = rep(1:3, each = 10))
```

We do not use the crossitem argument because it is does not make sense with nested trees. Also, we do not use the items argument in either of the two examples because we want to analyze all 30 items.

The resulting data frames contain binary item responses (value), a factor for the items (item) referring to the exogenize argument items, a factor for the persons (person), for the endnodes (endnode), for the crossitems (crossitem), and also one binary numeric variable per internal node in the latent-variable tree (exo$s$) with s referring to the internal node (e.g., exo1 to exo5 in the MA(1) example).

The glmer *code*

The glmer code will again not be explained in detail here. In the following, the code is given for three different models for the linlatT data.

(1) The free-covariance integrative model: (Embretson's 1991 model for learning and change)

```
R> freecov <- glmer(value ~ 0 + crossitem + endnode +
+    (0 + exo1 + exo3 + exo5 | person), family = binomial, data = linlatT)
```

Embretson's model uses the same item parameters for all time points, which explains the crossitem + endnode part of the code. For the case one would have time-specific item parameters, this part should be replaced with crossitem:endnode.

(2) The free MA(1) model

```
R> freeMA1 <- glmer(value ~ 0 + crossitem + endnode + (0 + exo1 | person) +
+     (0 + exo2 | person) + (0 + exo3 | person) + (0 + exo4 | person) +
+     (0 + exo5 | person), family = binomial, data = linlatT)
```

(3) The linear growth model with free MA(1) component

```
R> linlatT$time <- as.numeric(linlatT$endnode) - 1
R> linGfreeMA1 <- glmer(value ~ 0 + crossitem + time + (time | person) +
+    (0 + exo1 | person) + (0 + exo2 | person) + (0 + exo3 | person) +
+    (0 + exo4 | person) + (0 + exo5 | person),
+    family = binomial, data = linlatT)
```

If one wants to use the linear growth model without serial dependency, then the terms (0 + exo2 | person) and (0 + exo4 | person) should be omitted. Note that with these terms included there are actually seven components for a variance-covariance matrix having only six elements. This is due to the small example. From four time points on this problem does not occur.

For the nested tree model, only one model is listed:

(4) The bi-factor model

```
R> bifactor <- glmer(value ~ 0 + item + (0 + exo1 | person) +
+   (0 + exo2 | person) + (0 + exo3|  person) +
+   (0 + exo4 | person), family = binomial, data = neslatT)
```

3.4. Latent-variable tree applications with real data

How much do patients change in psychotherapy?

The first application concerns a linear growth curve model with serial dependency. The data are responses from 185 patients to the Output Questionnaire (Lambert *et al.* 2004) during three consecutive sessions of psychotherapy. The patients are selected from a much larger group on the basis of the availability of data from the first three sessions of psychotherapy. For several of the patients more than one set of questionnaire responses was available per session. The questionnaire is multidimensional (e.g., Kim *et al.* 2010) and for the present application 11 items are selected from a cluster with similar loadings in a PCA solution. These items have in common that they express fear and stress. The responses were given on a five-point scale and are dichotomized to illustrate the longitudinal IRTree analysis.

The research issue is whether and how much the patients improve through the first three sessions. Two preliminary analyses were performed, one to assess the within-patient variance per session, and one to assess the mean change through the sessions. These preliminary analyses are based on a model with one latent variable per session and an unconstrained covariance matrix, which is the saturated model for the latent structure. A multilevel approach was followed for the sometimes multiple responses per pair of patients and sessions. It was found that the within-patient variance per session did not contribute to the goodness of fit, and second, that free means per session did not contribute to the goodness of fit when compared with the fixed effect of a linear session covariate ($X_r = r - 1$).

In the further analysis, two linear growth models with residual components are compared: Model 1 without, and Model 2 with a free MA(1) component. In terms of the free MA(1) mapping matrix for $R = 3$, the first model uses only the internal nodes 1, 3, and 5, whereas the second uses also the internal nodes 2 and 4, in order to accomodate serial dependence. Assuming that the data frame is called `stressT` (not available in **irtrees**), the `glmer` code for Model 1 is:

```
R> linearplusres <- glmer(value ~ 0 + crossitem + time + (time | person) +
+    (0 + exo1 | person) + (0 + exo3 | person) + (0 + exo5 | person),
+    family = binomial, data = stressT)
```

The term `(time | person)` generates a random intercept θ_{p_0} and a random slope $\theta_{p,\mathrm{lin}}$ such that also their covariance is estimated. The three `exo` terms are the independent and time-point specific residuals (internal nodes 1, 3, 5).

Model 2 includes also a free MA(1) process and needs therefore also internal nodes 2 and 4. The `glmer` code is:

```
R> linearplusMA1 <- glmer(value ~ 0 + crossitem + time + (time | person) +
+    (0 + exo1 | person) + (0 + exo2 | person) + (0 + exo3 | person) +
+    (0 + exo4 | person) + (0 + exo5 | person), family = binomial,
+    data = stressT)
```

The log-likelihoods of the two models are −3299 (Model 1) and −3298 (Model 2). Given the small difference, the AIC and the BIC of the first model are better. In fact, also without the residual terms of Model 1, a log-likelihood of −3299 is obtained. Apparently the residuals and their possible dependence are not required for a good fit. This is perhaps not surprising since the model has already five parameters for six elements of the variance-covariance matrix, and Model 2 has even seven. The model with only time-specific latent variables but with free covariances has also a log-likelihood of −3298. Nevertheless we have calculated the two MA-weights (α_{12} and α_{23}) from the internal-node variances, for illustrative reasons. The internal node variance estimates are $\sigma^2_{\theta'_1} = 7.90e - 10$, $\sigma^2_{\theta'_2} = 2.89e - 02$, $\sigma^2_{\theta'_3} = 8.12e - 12$, $\sigma^2_{\theta'_4} = 2.28e - 01$, $\sigma^2_{\theta'_5} = 3.75e - 09$. The dependency variances (of θ'_2 and θ'_4) are much larger than the other variances. The estimated MA weights are 0.99983 (α_{12}) and 0.99994 (α_{23}), suggesting that if there is still some covariance unexplained by the random intercept plus random slope model, it is due to serial dependence of an integrative kind (given the values of α are almost 1.00). The estimates for the random intercept (θ_{p0}) variance, the random slope ($\theta_{p,\text{lin}}$) variance, and their correlation are 2.03, 0.176, 0.312, respectively, and the estimate of the fixed slope is −0.262 (se = 0.054). The patients seem to improve steadily, to a somewhat different extent depending on the patient, but without serious indications of serial dependency.

An overall measure of verbal aggression?

The second application concerns again the verbal aggression data `VerbAgg` included in **lme4**, and a bi-factor model is fit to the data, taking into account the three different verbally aggressive behaviors: cursing, scolding, and shouting. Each behavior is represented by eight items. The data are the dichotomized responses of 316 students to 24 items and the dichotomization corresponds to the top node in the linear response tree for the same data.

Three models will be considered. Model 1 is a one-dimensional model, with verbal aggression tendency as a common latent trait. Model 2 is one with only behavior-specific latent variables, but with free covariances, so that it is saturated for the structural part of the model. Model 3 is the earlier described bi-factor model, with a common verbal aggression tendency (top internal node), but with also behavior-specific latent variables, one for curse, one for scold, and one for shout (bottom internal nodes).

Package **irtrees** includes a data matrix `VerbAgg2` with the dichotomized responses ("no" vs. "perhaps" and "yes"). It can be exogenized for the IRTree analysis with a nested latent-variable tree as follows:

```
R> map <- cbind(c(1, 1, 1), c(1, 0, 0), c(0, 1, 0), c(0, 0, 1))
R> VerbAgg2T <- exogenize(VerbAgg2[, -(1:2)], map, endnode = rep(1:3, 8))
R> VerbAgg2T$value <- factor(VerbAgg2T$value)
```

The `glmer` code is as follows. Model 1:

```
R> onedim <- glmer(value ~ 0 + item + (1 | person),
+    family = binomial, data = VerbAgg2T)
```

Model 2:

```
R> threedim <- glmer(value ~ 0 + item + (0 + exo2 + exo3 + exo4 | person),
+    family = binomial, data = VerbAgg2T)
```

Model 3:

```
R> bifactor <- glmer(value ~ 0 + item + (0 + exo1 | person) +
+    (0 + exo2 | person) + (0 + exo3 | person) + (0 + exo4 | person),
+    family = binomial, data = VerbAgg2T)
```

Model 2 (log-likelihood = −3911, AIC = 7882, BIC = 8090) and Model 3 (log-likelihood = −3916, AIC = 7889, BIC = 8083) have a clearly better goodness of fit than Model 1 (log-likelihood = −4036, AIC = 8122, BIC = 8216). Models 2 and 3 are competitive. Model 2 is better on the AIC and Model 3 is better on the BIC. One may conclude from these results that the bi-factor model explains reasonably well the correlation between the three behaviors, such that the general latent trait can be used as an appropriate overall measure. The four estimated variances of the bi-factor model are 2.10 ($\sigma^2_{\theta'_1}$, general trait), 0.57 ($\sigma^2_{\theta'_2}$, cursing), 0.63 ($\sigma^2_{\theta'_3}$, scolding), and 2.26 ($\sigma^2_{\theta'_4}$, shouting). The larger variance of shouting corresponds with the fact that shouting correlates less with the other two behaviors than the other two among one another.

4. Discussion and conclusion

The four subcategories of models discussed in this manuscript extend the set of commonly considered item response models and have the advantage that they are members of the GLMM family, so that they can be estimated with general purpose GLMM software. Unfortunately, the GLMM family and the software that is illustrated have also limitations. Two serious limitations of the GLMM family are: (1) Only the sequential approach for ordinal response categories can be used, but not the partial-credit or graded-response approaches. (2) The two- and three-parameter models are excluded. There is also a possible issue with respect to the estimation of the models with the `glmer` function. Either the Laplace approximation is used, with a possible shrinkage of the variance estimates, or adaptive Gaussian quadrature is used, while for a bench mark data set, the difference with the Laplace approximation was only minor (also with a high number of quadrature points) and the shrinkage was not reduced. Another issue is that variance components cannot be easily tested on their significance (De Boeck *et al.* 2011). On the other hand, it is an important strength of the `glmer` function that it can easily cope with multilevel data and crossed random effects (persons and items), see (Doran *et al.* 2007). An alternative approach with a broader potential and which does not suffer in general from the underestimation of the variance components is a Bayesian approach for the same models. There is no conflict between such an approach and the models discussed here. Of course, also the usual IRT software packages can be used if they allow for item covariates. When a Gauss-Hermite approximation of the integral is implemented, these methods do not show the downward bias of the variance estimates (Tuerlinckx *et al.* 2004). The ambition of our manuscript is partly theoretical and partly practical. The theoretical results are relevant independent of the illustrated software.

The software package **irtrees** is a useful practical tool, but we hope to extend the `dendrify` function soon for the case of heterogenous items, with different numbers of categories, and for various types of data formats to start from. A more ambitious and far-reaching extension would be that the `exogenize` function is enriched with an algorithm that provides the user with the result of an appropriate exogenizing operation starting from a given structural equation model, on the condition that a tree can be constructed at all. Finally, we hope to add a function and example data for the combination of dendrification and exogenization, because the combination of response trees with latent-variable trees can make sense.

We hope that this manuscript can contribute to a perspective on item response models as a broader type of models than one commonly tends to believe, with relevance to a broader range of data sets and disciplines, and that it also contributes to the inclusion of item response models in the common families of statistical modeling approaches.

References

Andrich D (1978). "A Rating Formulation for Ordered Response Categories." *Psychometrika*, **43**, 561–573.

Batchelder W, Riefer D (1999). "Theoretical and Empirical Review of Multinomial Process Tree Modeling." *Psychonomic Bulletin & Review*, **6**, 57–86.

Bates D, Mächler M, Bolker B (2011). ***lme4**: Linear Mixed-Effects Models Using S4 Classes.* R package version 0.999375-42, URL http://CRAN.R-project.org/package=lme4.

Bock RD (1972). "Estimating Item Parameters and Latent Ability When Responses Are Scored in Two or More Mominal Categories." *Psychometrika*, **37**, 29–56.

Bradley RA, Terry ME (1952). "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons." *Biometrika*, **39**, 324–345.

Buss DM (1987). "Selection, Evocation, and Manipulation." *Journal of Personality and Social Psychology*, **53**, 1214–1221.

Cai L (2010). "A Two-Tier Full-Information Item Factor Analysis Model with Applications." *Psychometrika*, **75**, 581–612.

De Boeck P, Bakker M, Zwitser R, Nivard M, Hofman A, Tuerlinckx F, Partchev I (2011). "The Estimation of Item Response Models with the `lmer` Function from the **lme4** Package in R." *Journal of Statistical Software*, **39**(12), 1–28. URL http://www.jstatsoft.org/v39/i12/.

De Boeck P, Wilson M (2004). *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*. Springer-Verlag, New York.

Doran H, Bates D, Bliese P, Dowling M (2007). "Estimating the Multilevel Rasch Model: With the **lme4** Package." *Journal of Statistical Software*, **20**(2), 1–18. URL http://www.jstatsoft.org/v20/i02/.

Embretson S (1991). "A Multidimensional Latent Trait Model for Measuring Learning and Change." *Psychometrika*, **56**, 495–515.

Emmons EA, Diener E, Larsen RJ (1986). "Choice and Avoidance of Everyday Situations and Affect Congruence: Two Models of Reciprocal Interactionism." *Journal of Personality and Social Psychology*, **51**, 815–826.

Glas CA, Pimentel JL (2008). "Modeling Nonignorable Missing Data in Speeded Tests." *Educational and Psychological Measurement*, **68**, 907–922.

Greenwald AG, McGee DE, Schwartz JLK (1998). "Measuring Individual Differences in Implicit Cognition: The Implicit Association Test." *Journal of Personality and Social Psychology*, **74**, 1464–1480.

Gustafsson JE (1984). "A Unifying Model for the Structure of Intellectual Abilities." *Intelligence*, **8**, 179–203.

Hambleton RK, Swaminathan H, Rogers HJ (1991). *Fundamentals of Item Response Theory*. Sage Publications, Newbury Park.

Hemker B, van der Ark AL, Sijtsma K (2001). "On Measurement Properties of Continuation Ratio Models." *Psychometrika*, **66**, 487–506.

Holman R, Glas CAW (2005). "Modelling Non-Ignorable Missing-Data Mechanism with Item Response Theory Models." *British Journal of Mathematical and Statistical Psychology*, **58**, 1–17.

Hung LF (2010). "The Multigroup Multilevel Categorical Latent Growth Curve Models." *Multivariate Behavioral Research*, **45**, 359–392.

Joe H (2008). "Accuracy of Laplace approximation for Discrete Response Mixed Models." *Journal of Computational Statistics and Data Analysis*, **52**, 5066–5074.

Kim SH, Beretvas SN, Sherry AR (2010). "A Validation of the Factor Structure of OQ-45 Scores Using Factor Mixture Modeling." *Measurement and Evaluation in Counseling and Development*, **42**, 275–295.

Lambert MJ, Gregersen AT, Burlingname GM (2004). "The Outcome Questionnaire." In ME Maruish (ed.), *The Use of Psychological Testing for Treatment Planning and Outcome Assessment*, volume 3, 3rd edition, pp. 191–234. Lawrence Erlbaum, Mahwah.

Luce RD (1959). *Individual Choice Behavior*. John Wiley & Sons, Oxford.

MacLeod CM (1991). "Half a Century of Research on the Stroop Effect: An Integrative Review." *Psychological Bulletin*, **109**, 163–203.

Masters G (1982). "A Rasch Model for Partial Credit Scoring." *Psychometrika*, **47**, 149–174.

McArdle JJ, Grimm K, Hamagami F, Bowles R, Meredith W (2009). "Modeling Life-Span Growth Curves of Cognition Using Longitudinal Data with Multiple Samples and Changing Scales of Measurement." *Psychological Methods*, **14**, 126–149.

Partchev I, De Boeck P (2012). "Can Fast and Slow Intelligence Be Differentiated?" *Intelligence*, **40**, 23–32.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Revuelta J (2008). "The Generalized Logit-Linear Item Response Model for Binary-Designed Items." *Psychometrika*, **73**, 385–405.

Rijmen F (2010). "Formal Relations and an Empirical Comparison among the Bi-Factor, the Testlet, and a Second-Order Multidimensional IRT Model." *Journal of Educational Measurement*, **47**, 361–372.

Samejima F (1969). "Estimation of Latent Ability Using a Response Pattern of Graded Scores." *Psychometrika Monograph Supplement*, **34**.

Santos DMD, Berridge DM (2000). "A Continuation Ratio Random Effects Model for Repeated Ordinal Responses." *Statistics in Medicine*, **19**(24), 3377–3388.

Singer JD, Willett JB (1993). "It's About Time: Using Discrete-Time Survival Analysis to Study Duration and the Timing of Events." *Journal of Educational and Behavioral Statistics*, **18**, 155–195.

Steinberg L, Monahan KC (2007). "Age Differences in Resistence to Peer Influence." *Developmental Psychology*, **43**, 1531–1543.

Tuerlinckx F, Rijmen F, Molenberghs G, Verbeke G, Briggs D, Van den Noortgate W, Meulders M, De Boeck P (2004). "Estimation and Software." In P De Boeck, M Wilson (eds.), *Explanatory Item Response Models*. Springer-Verlag, New York.

Tutz G (1990). "Sequential Item Response Models with an Ordered Response." *British Journal of Mathematical and Statistical Psychology*, **43**, 39–55.

van der Ark LA (2001). "Relationships and Properties of Polytomous Item Response Models." *Applied Psychological Measurement*, **25**, 273–282.

Wang WC, Wu SL (2011). "The Random-Effect Generalized Rating Scale Model." *Journal of Educational Measurement*, **48**, 441–456.

Wickham H (2007). "Reshaping Data with the **reshape** Package." *Journal of Statistical Software*, **21**(12), 1–20. URL `http://www.jstatsoft.org/v21/i12/`.

Willett JB, Singer JD (1995). "It Is Déjá Vu All Over Again: Using Multiple-Spell Discrete-Time Survival Analysis." *Journal of Educational and Behavioral Statistics*, **20**, 41–67.

Wright SB, Matlen BJ, Baym CL, Ferrer E, Bunge SA (2008). "Neural Correlates of Fluid Reasoning in Children and Adults." *Frontiers in Human Neuroscience*, **1**(8), 1–8.

Affiliation:

Paul De Boeck
Department of Psychology
University of Amsterdam

Weesperplein 4
1018 XA Amsterdam, The Netherlands
E-mail: paul.deboeck@uva.nl
URL: http://www.fmg.uva.nl/psychologicalmethods/
and
Department of Psychology
KU Leuven
Tiensestraat 102
B-3000 Leuven, Belgium
URL: http://ppw.kuleuven.be/okp/home/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 48, Code Snippet 1 *Submitted:* 2011-09-16
May 2012 *Accepted:* 2012-04-14