Journal of Statistical Software

November 2006, Volume 17, Issue 5. http://www.jstatsoft.org/

ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses

Dimitris Rizopoulos
Catholic University of Leuven

Abstract

The R package **ltm** has been developed for the analysis of multivariate dichotomous and polytomous data using latent variable models, under the Item Response Theory approach. For dichotomous data the Rasch, the Two-Parameter Logistic, and Birnbaum's Three-Parameter models have been implemented, whereas for polytomous data Semejima's Graded Response model is available. Parameter estimates are obtained under marginal maximum likelihood using the Gauss-Hermite quadrature rule. The capabilities and features of the package are illustrated using two real data examples.

Keywords: latent variable models, item response theory, Rasch model, two-parameter logistic model, three-parameter model, graded response model.

1. Introduction

Latent variable models (Bartholomew and Knott 1999; Skrondal and Rabe-Hesketh 2004) constitute a general class of models suitable for the analysis of multivariate data. In principle, latent variable models are multivariate regression models that link continuous or categorical responses to unobserved covariates. The basic assumptions and objectives of latent variable modeling can be summarized as follows (Bartholomew, Steele, Moustaki, and Galbraith 2002):

- A small set of latent variables is assumed to explain the interrelationships in a set of observed response variables. This is known as the *conditional independence assumption*, which postulates that the response variables are independent given the latent variables. This simplifies the estimation procedure, since the likelihood contribution of the multivariate responses is decomposed into a product of independent terms. In addition, exploring conditional independence may help researchers first in drawing conclusions in complex situations, and second in summarizing the information from observed variables in few dimensions (reduction of dimensionality).

- Unobserved variables such as intelligence, mathematical or verbal ability, racial prejudice, political attitude, consumer preferences, which cannot be measured by conventional means, can be quantified by assuming latent variables. This is an attractive feature that has applications in areas such as educational testing, psychology, sociology, and marketing, in which such constructs play a very important role.

- Latent variable modeling is also used to assign scores to sample units in the latent dimensions based on their responses. This score (also known as a 'Factor Score') is a numerical value that indicates a person's relative spacing or standing on a latent variable. Factor scores may be used either to classify subjects or in the place of the original variables in a regression analysis, provided that the meaningful variation in the original data has not been lost.

Item Response Theory (IRT) (Baker and Kim 2004; van der Linden and Hambleton 1997) considers a class of latent variable models that link mainly dichotomous and polytomous manifest (i.e., response) variables to a single latent variable. The main applications of IRT can be found in educational testing in which analysts are interested in measuring examinees' ability using a test that consists of several items (i.e., questions). Several models and estimation procedures have been proposed that deal with various aspects of educational testing.

The aim of this paper is to present the R (R Development Core Team 2006) package **ltm**, available from CRAN (`http://CRAN.R-project.org/`), which can be used to fit a set of latent variable models under the IRT approach. The main focus of the package is on dichotomous and polytomous response data. For Gaussian manifest variables the function `factanal()` of package **stats** can be used.

The paper is organized as follows. Section 2 briefly reviews the latent variable models for dichotomous and polytomous data. In Section 3 the use of the main functions and methods of **ltm** is illustrated using two real examples. Finally, in Section 4 we describe some extra features of **ltm** and refer to future extensions.

2. Latent variable models formulation

The basic idea of latent variable analysis is to find, for a given set of response variables $x_1, \ldots, x_p$, a set of latent variables $z_1, \ldots, z_q$ (with $q \ll p$) that contains essentially the same information about dependence. The latent variable regression models have usually the following form

$$E(x_i \mid z) \quad = \quad g(\lambda_{i0} + \lambda_{i1} z_1 + \cdots + \lambda_{iq} z_q) \qquad (i = 1, \ldots, p), \tag{1}$$

where $g(\cdot)$ is a link function, $\lambda_{i0}, \ldots, \lambda_{iq}$ are the regression coefficients for the ith manifest variable, and x_i is independent of x_j, for $i \neq j$, given $z = \{z_1, \ldots, z_q\}$. The common factor analysis model assumes that the x_i's are continuous random variables following a Normal distribution with $g(\cdot)$ being the identity link. In this paper we focus on IRT models, and consider mainly dichotomous and polytomous items, in which $E(x_i \mid z)$ expresses the probability of endorsing one of the possible response categories. In the IRT framework usually one latent variable is assumed, but for models on dichotomous responses the inclusion of two latent variables is briefly discussed in Section 4.

2.1. Models for dichotomous data

The basic ingredient of the IRT modeling for dichotomous data is the model for the probability of positive (or correct) response in each item given the ability level z. A general model for this probability for the mth examinee in the ith item is the following

$$P(x_{im} = 1 \mid z_m) \quad = \quad c_i + (1 - c_i)g\{\alpha_i(z_m - \beta_i)\}, \tag{2}$$

where x_{im} is the dichotomous manifest variable, z_m denotes the examinee's level on the latent scale, c_i is the guessing parameter, α_i the discrimination parameter and β_i the difficulty parameter. The guessing parameter expresses the probability that an examinee with very low ability responds correctly to an item by chance. The discrimination parameter quantifies how well the item distinguishes between subjects with low/high standing in the latent scale, and the difficulty parameter expresses the difficulty level of the item.

The one-parameter logistic model, also known as the Rasch model (Rasch 1960), assumes that there is no guessing parameter, i.e., $c_i = 0$ and that the discrimination parameter equals one, i.e., $\alpha_i = 1$, $\forall i$. The two-parameter logistic model allows for different discrimination parameters per item and assumes that $c_i = 0$. Finally, Birnbaum's three-parameter model (Birnbaum 1968) estimates all three parameters per item.

The two most common choices for $g(\cdot)$ are the probit and the logit link, which correspond to the cumulative distribution function (cdf) of the normal and logistic distributions, respectively. The functions included in **ltm** fit (2) under the logit link. Approximate results under the probit link for the one- and two-parameter logistic models can be obtained using the relation

$$\alpha_i^{(p)}(z_m - \beta_i^{(p)}) \quad \approx \quad 1.702\alpha_i^{(l)}(z_m - \beta_i^{(l)}), \tag{3}$$

where $\alpha_i^{(p)}$, $\alpha_i^{(l)}$ are the discrimination parameters under the probit and logit link, respectively, and $\beta_i^{(p)}$, $\beta_i^{(l)}$ are defined analogously. The scaling constant 1.702 is chosen such that the absolute difference between the normal and logistic cdf is less than 0.01 over the real line.

2.2. Models for polytomous ordinal data

Analysis of polytomous manifest variables is currently handled by **ltm** using the Graded Response Model (GRM). The GRM was first introduced by Samejima (1969), and postulates that the probability of the mth subject to endorse the kth response for the ith item is expressed as

$$P(x_{im} = k \mid z_m) \quad = \quad g(\eta_{ik}) - g(\eta_{i,k+1}), \tag{4}$$

$$\eta_{ik} \quad = \quad \alpha_i(z_m - \beta_{ik}), \quad k = 1, \ldots, K_i,$$

where x_{im} is the ordinal manifest variable with K_i possible response categories, z_m is the standing of the mth subject in the latent trait continuum, α_i denotes the discrimination parameter, and β_{ik}'s are the extremity parameters with $\beta_{i1} < \ldots < \beta_{ik} < \ldots < \beta_{i,K_i-1}$ and $\beta_{iK_i} = \infty$. The interpretation of α_i is essentially the same as in the models for dichotomous data. However, in GRM the β_{ik}'s represent the cut-off points in the cumulative probabilities scale and thus their interpretation is not direct. **ltm** fits the GRM under the logit link.

There have been proposed several alternatives to the GRM for the analysis of polytomously scored items. Two of them that are frequently applied are the Partial Credit and the Rating Scale models. The partial credit model is more suitable in cases where the difference between response options is identical for different items in the attitude scale, whereas the rating scale model is applicable to a test in which all items have the same number of categories. We refer to van der Linden and Hambleton (1997) and Zickar (2002) for additional information and discussion about the polytomous models.

2.3. Implementation in ltm

Estimation of model parameters has received a lot of attention in the IRT literature. Under Maximum Likelihood there have been developed three major methods, namely conditional, full, and marginal maximum likelihood. A detailed overview of these methods is presented in Baker and Kim (2004) and a brief discussion about the relative merits of each method can be found in Agresti (2002, Section 12.1.5). In addition, parameter and ability estimation under a Bayesian approach is reviewed in Baker and Kim (2004). Package **ltm** fits the models presented in Sections 2.1 and 2.2 using Marginal Maximum Likelihood Estimation (MMLE). Conditional maximum likelihood estimation has been recently implemented in package **eRm** (Mair and Hatzinger 2006) but only for some Rasch type models, and Markov Chain Monte Carlo for the one and k dimensional latent trait models are available from the **MCMCpack** package (Martin and Quinn 2006).

Parameter estimation under MMLE assumes that the respondents represent a random sample from a population and their ability is distributed according to a distribution function $F(z)$. The model parameters are estimated by maximizing the observed data log-likelihood obtained by integrating out the latent variables; the contribution of the mth sample unit is

$$\ell_m(\theta) \quad = \quad \log p(x_m;\theta) = \log \int p(x_m|z_m;\theta)\, p(z_m)\, dz_m, \tag{5}$$

where $p(\cdot)$ denotes a probability density function, x_m denotes the vector of responses for the mth sample unit, z_m is assumed to follow a standard normal distribution and $\theta = (\alpha_i, \beta_i)$. Package **ltm** contains four model fitting functions, namely `rasch()`, `ltm()`, `tpm()` and `grm()` for fitting the Rasch model, the latent trait model, the three-parameter model, and the graded response model, respectively. The latent trait model is a general latent variable model for dichotomous data of the form (1), including as a special case the two-parameter logistic model. The integral in (5) is approximated using the Gauss-Hermite quadrature rule. By default, in `rasch()`, `tpm()` and `grm()` 21 quadrature points are used, whereas `ltm()` uses 21 points when one latent variable is specified and 15 otherwise. It is known (Pinheiro and Bates 1995) that the number of quadrature points used may influence the parameter estimates, standard errors and log-likelihood value, especially for the case of two latent variables and nonlinear terms as described in Section 4. Thus, it is advisable to investigate its effect by fitting the model with an increasing number of quadrature points. However, for the unidimensional (i.e., one latent variable) IRT models considered so far, the default number of points will be, in the majority of the cases, sufficient.

Maximization of the integrated log-likelihood (5) with respect to θ for `rasch()`, `tpm()` and `grm()` is achieved using `optim()`'s BFGS algorithm. For `ltm()` a hybrid algorithm is adopted, in which a number of EM iterations is initially used, followed by BFGS iterations until convergence. In addition, for all four functions, the optimization procedure works under an

additive parameterization as in (1), i.e., $\lambda_{i0} + \lambda_{i1} z_m$; however, the parameter estimates for the Rasch, the two-parameter logistic, the three-parameter, and the graded response models are returned, by default, under parameterizations (2) and (4). This feature is controlled by the `IRT.param` argument. Starting values are obtained either by fitting univariate GLMs to the observed data with random or deterministic z values, or they can be explicitly set using the `start.val` argument. The option of random starting values (i.e., use of random z values in the univariate GLM) might be useful for revealing potential local maxima issues. By default all functions use deterministic starting values (i.e., use of deterministic z values in the univariate GLM). Furthermore, all four functions have a `control` argument that can be used to specify various control values, such as the optimization method in `optim()` (for `tpm()` the `nlminb()` optimizer is also available) and the corresponding maximum number of iterations, and the number of quadrature points, among others. Finally, the four fitting functions return objects of class named after the corresponding (model fitting) function (i.e., `rasch()` returns `rasch` objects, etc.), for which the following methods are available: `print()`, `coef()`, `summary()`, `plot()`, `fitted()`, `vcov()`, `logLik()`, `anova()`, `margins()` and `factor.scores()`; the last two generic functions are defined in **ltm** and their use is illustrated in more detail in the following section.

3. Package ltm in use

We shall demonstrate the use of **ltm** in two data sets; the first one concerns binary data where `rasch()`, `ltm()` and `tpm()`, and their methods are investigated, while for the second one that deals with ordinal data, `grm()` and its methods are illustrated. For both examples the results are presented under the default number of quadrature points. To investigate sensitivity we have also fitted the models with 61 points and essentially the same results have been obtained.

3.1. An example with binary data

In this section we consider data from the Law School Admission Test (LSAT) that has been taken by 1000 individuals responding to five questions. This is a typical example of an educational test data-set presented also in Bock and Lieberman (1970). LSAT data are available in **ltm** as the `data.frame` `LSAT`.

At an initial step, descriptive statistics for LSAT are produced using the `descript()` function:

```
R> descript(LSAT)

Descriptive statistics for 'LSAT' data-set

Sample:
 5 items and 1000 sample units; 0 missing values

Proportions for each level of response:
      Item 1 Item 2 Item 3 Item 4 Item 5
0      0.076  0.291  0.447  0.237  0.130
1      0.924  0.709  0.553  0.763  0.870
logit  2.498  0.891  0.213  1.169  1.901
```

```
Frequencies of total scores:
     0  1  2   3   4   5
Freq 3 20 85 237 357 298


Biserial correlation with Total Score:
Item 1 Item 2 Item 3 Item 4 Item 5
 0.362  0.566  0.618  0.534  0.435


Pairwise Associations:
   Item i Item j p.value
1       1      5   0.565
2       1      4   0.208
3       3      5   0.113
4       2      4   0.059
5       1      2   0.028
6       2      5   0.009
7       1      3   0.003
8       4      5   0.002
9       3      4   7e-04
10      2      3   4e-04
```

The output of `descript()` contains among others the χ^2 p-values for pairwise associations between the five items, corresponding to the 2×2 contingency tables for all possible pairs. Inspection of non significant results can be used to reveal 'problematic' items[1]. In addition, for the LSAT data we observe that item 1 seems to be the easiest one having the highest proportion of correct responses, while only three pairs of items seem to have low degree of association.

We initially fit the original form of the Rasch models that assumes known discrimination parameter fixed at one. The version of the Rasch model fitted by `rasch()` in **ltm** assumes equal discrimination parameters across items but by default estimates its value, i.e., for p items $\alpha_1 = \ldots = \alpha_p = \alpha$. In order to impose the constraint $\alpha = 1$, the `constraint` argument is used. This argument accepts a two-column matrix where the first column denotes the parameter and the second column indicates the value at which the corresponding parameter should be fixed. Parameters are fixed under the additive parameterization $\lambda_{i0} + \lambda z_m$; for instance, for p items the numbers $1, \ldots, p$, in the first column of `constraint`, correspond to parameters $\lambda_{10}, \ldots \lambda_{p0}$, and the number $p + 1$ to the discrimination parameter λ.[2] Thus, for the LSAT data-set we fix the discrimination parameter at one by:

```
R> fit1 <- rasch(LSAT, constraint = cbind(length(LSAT) + 1, 1))
R> summary(fit1)
```

[1] Latent variable models assume that the high associations between items can be explained by a set of latent variables. Thus, for pairs of items that do not reject independence we could say that they violate this assumption.

[2] Note that under both parameterizations, the discrimination parameter coincides, i.e., $\lambda \equiv \alpha$.

```
Call:
rasch(data = LSAT, constraint = cbind(length(LSAT) + 1, 1))

Model Summary:
   log.Lik      AIC      BIC
 -2473.054 4956.108 4980.646

Coefficients:
             value std.err   z.vals
Dffclt.It1 -2.8720  0.1287 -22.3066
Dffclt.It2 -1.0630  0.0821 -12.9458
Dffclt.It3 -0.2576  0.0766  -3.3635
Dffclt.It4 -1.3881  0.0865 -16.0478
Dffclt.It5 -2.2188  0.1048 -21.1660
Dscrmn      1.0000      NA       NA

Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 6.3e-05
quasi-Newton: BFGS
```

The results of the descriptive analysis are also validated by the model fit, where items 3 and 1 are the most difficult and the easiest, respectively. The parameter estimates can be transformed to probability estimates using the `coef()` method:

```
R> coef(fit1, prob = TRUE, order = TRUE)

       Dffclt Dscrmn P(x=1|z=0)
Item 1 -2.872      1      0.946
Item 5 -2.219      1      0.902
Item 4 -1.388      1      0.800
Item 2 -1.063      1      0.743
Item 3 -0.258      1      0.564
```

The column `P(x=1|z=0)` corresponds to $P(x_i = 1 \mid z = 0)$ under (2), and denotes the probability of a positive response to the ith item for the average individual. The `order` argument can be used to sort the items according to the difficulty estimates.

In order to check the fit of the model to the data, the `GoF.rasch()` and `margins()` functions are used. The `GoF.rasch()` function performs a parametric Bootstrap goodness-of-fit test using Pearson's χ^2 statistic. In particular, the null hypothesis states that the observed data have been generated under the Rasch model with parameter values the maximum likelihood estimates $\hat{\theta}$. To test this hypothesis B samples are generated under the Rasch model using $\hat{\theta}$, and the Pearson's χ^2 statistic T_b $(b = 1, \ldots, B)$ is computed for each data-set; the p-value is

then approximated by the number of times $T_b \geq T_{obs}$ plus one, divided by $B + 1$, where T_{obs} denotes the value of the statistic in the original data-set. For the LSAT data this procedure yields:

```
R> GoF.rasch(fit1, B = 199)

Goodness-of-Fit using Pearson chi-squared

Call: rasch(data = LSAT, constraint = cbind(length(LSAT) + 1, 1))

Tobs: 30.6
# Bootstrap samples: 200
p-value: 0.235
```

Based on 200 data-sets, the non significant p-value suggests an acceptable fit of the model. An alternative method to investigate the fit of the model is to examine the two- and three-way χ^2 residuals produced by the `margins()` method:

```
R> margins(fit1)

Call:
rasch(data = LSAT, constraint = cbind(length(LSAT) + 1, 1))

Fit on the Two-Way Margins

Response: (0,0)
  Item i Item j Obs   Exp (O-E)^2/E
1      2      4  81 98.69      3.17
2      1      5  12 18.45      2.25
3      3      5  67 80.04      2.12

Response: (1,0)
  Item i Item j Obs    Exp (O-E)^2/E
1      3      5  63  51.62      2.51
2      2      4 156 139.78      1.88
3      3      4 108  99.42      0.74

Response: (0,1)
  Item i Item j Obs    Exp (O-E)^2/E
1      2      4 210 193.47      1.41
2      2      3 135 125.07      0.79
3      1      4  53  47.24      0.70

Response: (1,1)
  Item i Item j Obs    Exp (O-E)^2/E
1      2      4 553 568.06      0.40
2      3      5 490 501.43      0.26
3      2      3 418 427.98      0.23
```

These residuals are calculated by constructing all possible 2×2 contingency tables for the available items and checking the model fit in each cell using the Pearson's χ^2 statistic. The `print()` method for class `margins` returns the pairs or triplets of items with the three (this number is controlled by the `nprint` argument) highest residual values for each combinations of responses (i.e., for each cell of the contingency table). Using as a rule of thumb the value 3.5, we observe that the constrained version of the Rasch model provides a good fit to the two-way margins. We continue by examining the fit to the three-way margins, which are constructed analogously:

```
R> margins(fit1, type = "three-way", nprint = 2)

Call:
rasch(data = LSAT, constraint = cbind(length(LSAT) + 1, 1))

Fit on the Three-Way Margins

Response: (0,0,0)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      2      3      4  48 66.07      4.94 ***
2      1      3      5   6 13.58      4.23 ***

Response: (1,0,0)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      1      2      4  70 82.01      1.76
2      2      4      5  28 22.75      1.21

Response: (0,1,0)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      1      2      5   3  7.73      2.90
2      3      4      5  37 45.58      1.61

Response: (1,1,0)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      3      4      5  48  36.91      3.33
2      1      2      4 144 126.35      2.47

Response: (0,0,1)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      1      3      5  41 34.58      1.19
2      2      4      5  64 72.26      0.94

Response: (1,0,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      1      2      4 190 174.87      1.31
2      1      2      3 126 114.66      1.12
```

```
Response: (0,1,1)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      1      2      5  42 34.35      1.70
2      1      4      5  46 38.23      1.58

Response: (1,1,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      3      4      5 397 416.73      0.93
2      2      3      4 343 361.18      0.91

'***' denotes a chi-squared residual greater than 3.5
```

The three-way margins suggest a problematic fit for two triplets of items, both containing item 3.

We shall continue by fitting the unconstrained version of the Rasch model, which can be done by calling `rasch()` without specifying the `constraint` argument:

```
R> fit2 <- rasch(LSAT)
R> summary(fit2)

Call:
rasch(data = LSAT)

Model Summary:
   log.Lik      AIC      BIC
 -2466.938 4945.875 4975.322

Coefficients:
             value std.err   z.vals
Dffclt.It1 -3.6153  0.3266 -11.0680
Dffclt.It2 -1.3224  0.1422  -9.3009
Dffclt.It3 -0.3176  0.0977  -3.2518
Dffclt.It4 -1.7301  0.1691 -10.2290
Dffclt.It5 -2.7802  0.2510 -11.0743
Dscrmn      0.7551  0.0694  10.8757

Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 2.5e-05
quasi-Newton: BFGS
```

The output suggests that the discrimination parameter is different from 1. This can be formally tested via a likelihood ratio test using `anova()`:

```
R> anova(fit1, fit2)

 Likelihood Ratio Table
         AIC     BIC  log.Lik   LRT df p.value
fit1 4956.11 4980.65 -2473.05
fit2 4945.88 4975.32 -2466.94 12.23  1  <0.001
```

The LRT verifies that the unconstrained version of the Rasch model is more suitable for the LSAT data. The definitions of AIC and BIC used by the `summary()` and `anova()` methods in **ltm** are such that "smaller is better". The same conclusion is also supported by examining the fit of the unconstrained model to the three-way margins in which all residuals have acceptable values:

```
R> margins(fit2, type = "three-way", nprint = 2)

Call:
rasch(data = LSAT)

Fit on the Three-Way Margins

Response: (0,0,0)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      1      3      5   6  9.40      1.23
2      3      4      5  30 25.85      0.67

Response: (1,0,0)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      2      4      5  28 22.75      1.21
2      2      3      4  81 74.44      0.58

Response: (0,1,0)
  Item i Item j Item k Obs  Exp (O-E)^2/E
1      1      2      5   3 7.58      2.76
2      1      3      4   5 9.21      1.92

Response: (1,1,0)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      2      4      5  51 57.49      0.73
2      3      4      5  48 42.75      0.64

Response: (0,0,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      1      3      5  41  33.07      1.90
2      2      3      4 108 101.28      0.45
```

```
Response: (1,0,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      2      3      4 210 218.91      0.36
2      1      2      4 190 185.56      0.11

Response: (0,1,1)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      1      3      5  23 28.38      1.02
2      1      4      5  46 42.51      0.29

Response: (1,1,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      1      2      4 520 526.36      0.08
2      1      2      3 398 393.30      0.06
```

Finally, we investigate two possible extensions of the unconstrained Rasch model. First, we test if the two-parameter logistic model, which assumes a different discrimination parameter per item, provides a better fit than the unconstrained Rasch model. The two-parameter logistic model can be fitted using `ltm()`. In particular, `ltm()` accepts as first argument an R `formula`, in which its left-hand side must be the `data.frame` or `matrix` of dichotomous responses and the right-hand side specifies the latent structure. For the latent structure, up to two latent variables are allowed with code names `z1` and `z2`. The two-parameter logistic model for the LSAT data can be fitted and compared with the unconstrained Rasch model as follows:

```
R> fit3 <- ltm(LSAT ~ z1)
R> anova(fit2, fit3)

 Likelihood Ratio Table
         AIC     BIC  log.Lik  LRT df p.value
fit2 4945.88 4975.32 -2466.94
fit3 4953.31 5002.38 -2466.65 0.57  4   0.967
```

Second, we test whether incorporating a guessing parameter to the unconstrained Rasch model improves the fit. This extension can be fitted using `tpm()`, which has syntax very similar to `rasch()` and allows one to fit either a Rasch model with a guessing parameter or the three-parameter model as described in Section 2.1. To fit the unconstrained Rasch model with a guessing parameter the `type` argument needs to be specified as

```
R> fit4 <- tpm(LSAT, type = "rasch", max.guessing = 1)
R> anova(fit2, fit4)

 Likelihood Ratio Table
         AIC     BIC  log.Lik  LRT df p.value
fit2 4945.88 4975.32 -2466.94
fit4 4955.46 5009.45 -2466.73 0.41  5   0.995
```

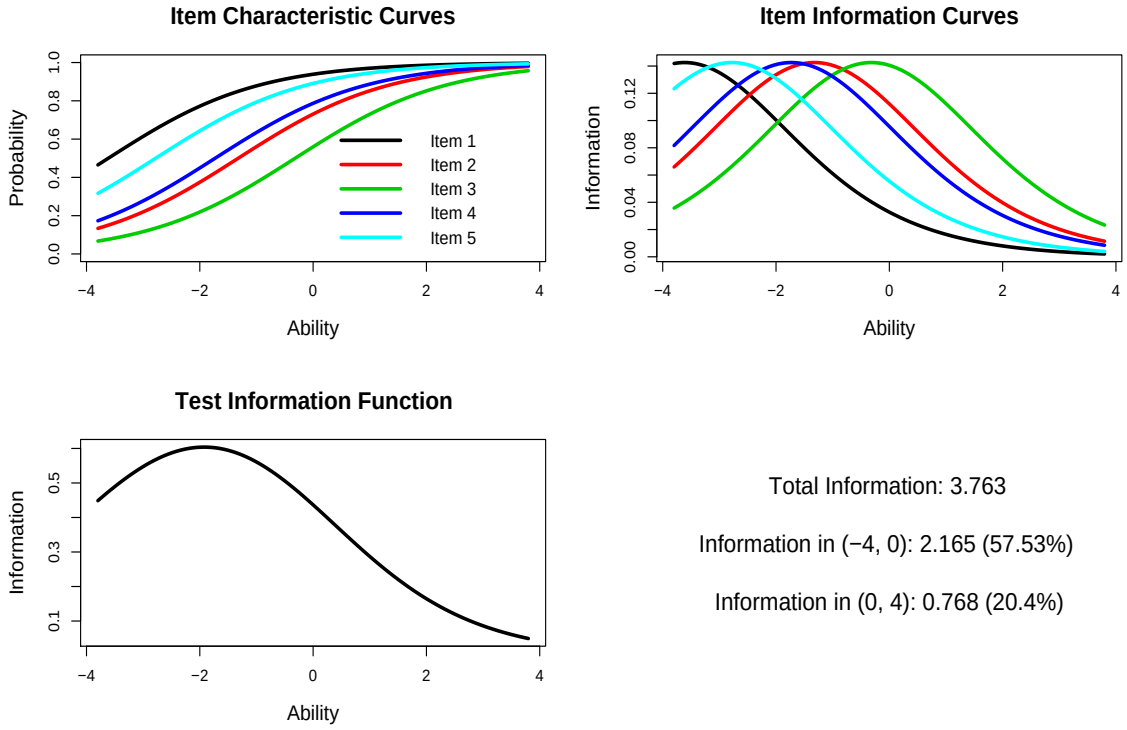

Figure 1: Item Characteristic, Item Information and Test Information Curves for the LSAT data-set under the unconstrained Rasch model.

The `max.guessing` argument specifies the upper bound for the guessing parameters. For both extensions the data clearly suggest that they are not required. The same conclusion is supported by the AIC and BIC values as well. Adopting the unconstrained Rasch model as the more appropriate for the LSAT data, we produce the Item Characteristic, the Item Information and the Test Information Curves, by appropriate calls to the `plot()` method for class `rasch`. All the plots are combined in Figure 1. The R code used to produce this figure can be found in Appendix A. The utility function `information()`, used in the figure, computes the area under the Test or Item Information Curves in a specified interval. According to the Test Information Curve we observe that the items asked in LSAT mainly provide information for respondents with low ability. In particular, the amount of test information for ability levels in the interval $(-4, 0)$ is almost 60%, whereas the item that seems to distinguish between respondents with higher ability levels is the third one.

Finally, the ability estimates can be obtained using the `factor.scores()` function (for more details regarding factor scores estimation see Section 4):

```
R> factor.scores(fit2)

Call:
rasch(data = LSAT)
```

```
Scoring Method: Empirical Bayes

Factor-Scores for observed response patterns:
   Item 1 Item 2 Item 3 Item 4 Item 5 Obs     Exp     z1 se.z1
1       0      0      0      0      0   3   2.364 -1.910 0.790
2       0      0      0      0      1   6   5.468 -1.439 0.793
3       0      0      0      1      0   2   2.474 -1.439 0.793
4       0      0      0      1      1  11   8.249 -0.959 0.801
5       0      0      1      0      0   1   0.852 -1.439 0.793
6       0      0      1      0      1   1   2.839 -0.959 0.801
7       0      0      1      1      0   3   1.285 -0.959 0.801
8       0      0      1      1      1   4   6.222 -0.466 0.816
9       0      1      0      0      0   1   1.819 -1.439 0.793
10      0      1      0      0      1   8   6.063 -0.959 0.801
11      0      1      0      1      1  16  13.288 -0.466 0.816
12      0      1      1      0      1   3   4.574 -0.466 0.816
13      0      1      1      1      0   2   2.070 -0.466 0.816
14      0      1      1      1      1  15  14.749  0.049 0.836
15      1      0      0      0      0  10  10.273 -1.439 0.793
16      1      0      0      0      1  29  34.249 -0.959 0.801
17      1      0      0      1      0  14  15.498 -0.959 0.801
18      1      0      0      1      1  81  75.060 -0.466 0.816
19      1      0      1      0      0   3   5.334 -0.959 0.801
20      1      0      1      0      1  28  25.834 -0.466 0.816
21      1      0      1      1      0  15  11.690 -0.466 0.816
22      1      0      1      1      1  80  83.310  0.049 0.836
23      1      1      0      0      0  16  11.391 -0.959 0.801
24      1      1      0      0      1  56  55.171 -0.466 0.816
25      1      1      0      1      0  21  24.965 -0.466 0.816
26      1      1      0      1      1 173 177.918  0.049 0.836
27      1      1      1      0      0  11   8.592 -0.466 0.816
28      1      1      1      0      1  61  61.235  0.049 0.836
29      1      1      1      1      0  28  27.709  0.049 0.836
30      1      1      1      1      1 298 295.767  0.593 0.862
```

By default `factor.scores()` produces ability estimates for the observed response patterns; if ability estimates are required for non observed or specific response patterns, these could be specified using the `resp.patterns` argument, for instance:

```
R> factor.scores(fit2, resp.patterns = rbind(c(0,1,1,0,0), c(0,1,0,1,0)))

Call:
rasch(data = LSAT)

Scoring Method: Empirical Bayes
```

```
Factor-Scores for specified response patterns:
  Item 1 Item 2 Item 3 Item 4 Item 5 Obs   Exp     z1 se.z1
1      0      1      1      0      0   0 0.944 -0.959 0.801
2      0      1      0      1      0   0 2.744 -0.959 0.801
```

3.2. An example with ordinal data

The data we consider here come from the Environment section of the 1990 British Social Attitudes Survey (Brook, Taylor, and Prior 1991; Bartholomew *et al.* 2002). The data frame `Environment` available in **ltm** contains the responses of 291 individuals asked about their opinion on six environmental issues. The response options were "very concerned", "slightly concerned" and "not very concerned," thus giving rise to six ordinal items.

As for the LSAT data, the `descript()` function can be used to produce descriptive statistics for the Environment data-set (output not shown). We can observe that for all six items the first response level has the highest frequency, followed by the second and third levels. The p-values for the pairwise associations indicate significant associations between all items. An alternative method to explore the degree of association between pairs of items is the computation of a nonparametric correlation coefficient. The `rcor.test()` function provides this option:

```
R> rcor.test(Environment, method = "kendall")
```

```
             LeadPetrol RiverSea RadioWaste AirPollution Chemicals Nuclear
LeadPetrol   ****       0.385    0.260      0.457        0.305     0.279
RiverSea     < 0.001    ****     0.399      0.548        0.403     0.320
RadioWaste   < 0.001    < 0.001  ****       0.506        0.623     0.484
AirPollution < 0.001    < 0.001  < 0.001    ****         0.504     0.382
Chemicals    < 0.001    < 0.001  < 0.001    < 0.001      ****      0.463
Nuclear      < 0.001    < 0.001  < 0.001    < 0.001      < 0.001   ****

upper diagonal part contains correlation coefficient estimates
lower diagonal part contains corresponding p-values
```

The implementation of `rcor.test()` is based on the `cor()` function of package **stats** and thus it provides two options for nonparametric correlation coefficients, namely Kendall's *tau* and Spearman's *rho*, controlled by the `method` argument. The `print()` method for class `rcor.test` returns a square matrix in which the upper diagonal part contains the estimates of the correlation coefficients, and the lower diagonal part contains the corresponding p-values.

Initially, we fit the constrained version of the GRM that assumes equal discrimination parameters across items (i.e., $\alpha_i = \alpha$ for all i in (4)). This model could be considered as the equivalent of the Rasch model for ordinal data. The constrained GRM is fitted by `grm()` as follows:

```
R> fit1 <- grm(Environment, constrained = TRUE)
R> fit1
```

```
Call:
grm(data = Environment, constrained = TRUE)

Coefficients:
              Extrmt1  Extrmt2  Dscrmn
LeadPetrol      0.388    1.966   2.233
RiverSea        1.047    2.533   2.233
RadioWaste      0.820    1.975   2.233
AirPollution    0.475    2.420   2.233
Chemicals       0.844    2.025   2.233
Nuclear         0.056    1.251   2.233

Log.Lik: -1106.334
```

A more detailed output can be produced using the `summary()` method. This contains the AIC and BIC values, and extra information for the optimization procedure, as in the `summary()` method for `rasch` objects. If standard errors for the parameter estimates are required, these could be obtained by specifying `Hessian = TRUE` in the call to `grm()`.

The fit of the model can be checked using the `margins()` method for class `grm`. The two-way margins are obtained by:

```
R> margins(fit1)

Call:
grm(data = Environment, constrained = TRUE)

Fit on the Two-Way Margins

             LeadPetrol RiverSea RadioWaste AirPollution Chemicals Nuclear
LeadPetrol   -           9.82    10.12       5.05         7.84     17.10
RiverSea                -         5.03      16.51         2.55      7.17
RadioWaste                      -           6.50        20.37     11.74
AirPollution                              -             4.35      4.58
Chemicals                                               -         3.68
Nuclear                                                           -
```

The output includes a square matrix in which the upper diagonal part contains the residuals, and the lower diagonal part indicates the pairs for which the residuals exceed the threshold value. Analogously, the three-way margins are produced by:

```
R> margins(fit1, type = "three")

Call:
grm(data = Environment, constrained = TRUE)
```

```
Fit on the Three-Way Margins

   Item i Item j Item k (O-E)^2/E
1       1      2      3     28.34
2       1      2      4     33.36
3       1      2      5     29.59
4       1      2      6     42.48
5       1      3      4     32.46
6       1      3      5     66.27
7       1      3      6     64.81
8       1      4      5     25.10
9       1      4      6     34.45
10      1      5      6     39.31
11      2      3      4     28.79
12      2      3      5     37.33
13      2      3      6     32.07
14      2      4      5     26.28
15      2      4      6     36.16
16      2      5      6     19.22
17      3      4      5     38.63
18      3      4      6     26.33
19      3      5      6     39.08
20      4      5      6     22.00
```

Both the two- and three-way residuals show a good fit of the constrained model to the data. However, checking the fit of the model in the margins does not correspond to an overall goodness-of-fit test, thus the unconstrained GRM will be fitted as well:

```
R> fit2 <- grm(Environment)
R> fit2

Call:
grm(data = Environment)

Coefficients:
              Extrmt1  Extrmt2  Dscrmn
LeadPetrol      0.463    2.535   1.393
RiverSea        1.017    2.421   2.440
RadioWaste      0.755    1.747   3.157
AirPollution    0.440    2.131   3.141
Chemicals       0.788    1.835   2.900
Nuclear         0.054    1.378   1.811

Log.Lik: -1091.232
```

In order to check if the unconstrained GRM provides a better fit than the constrained GRM, a likelihood ratio test is used:

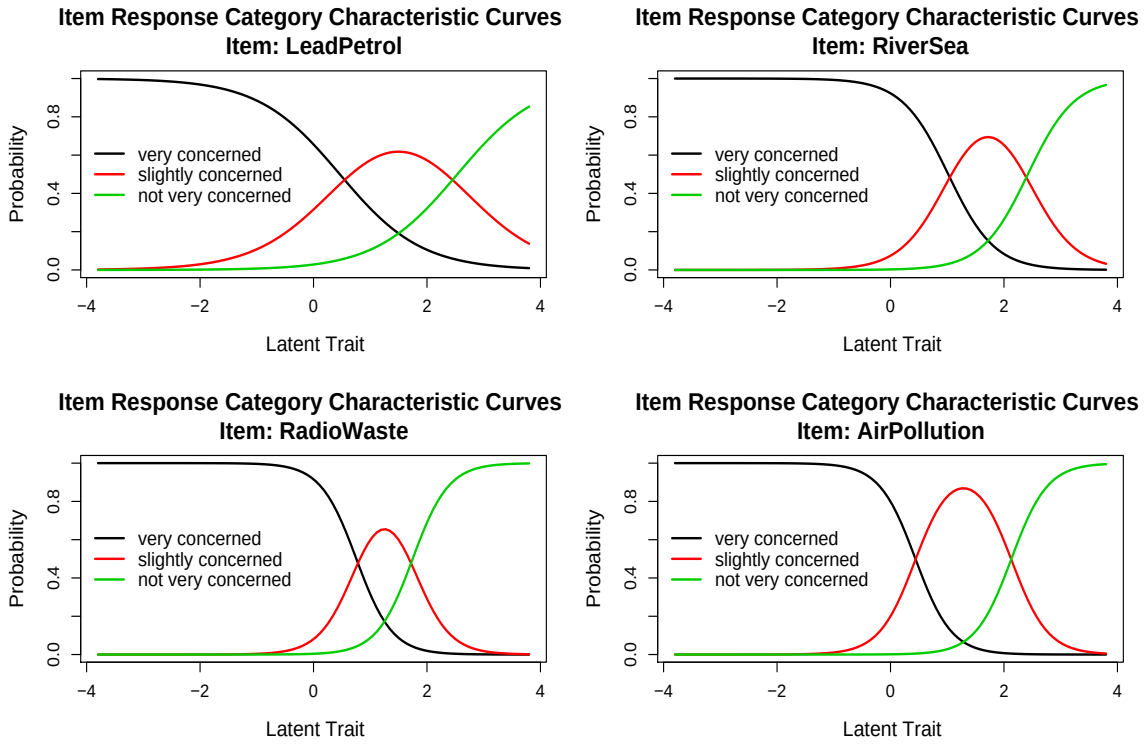

Figure 2: Item Characteristic Curves for the first 4 items, for the Environment data-set under the unconstrained GRM model.

```
R> anova(fit1, fit2)
```

```
 Likelihood Ratio Table
         AIC     BIC  log.Lik   LRT df p.value
fit1 2238.67 2286.42 -1106.33
fit2 2218.46 2284.58 -1091.23 30.21  5  <0.001
```

The LRT indicates that the unconstrained GRM is preferable for the Environment data.

The fitted unconstrained GRM is illustrated in Figures 2 and 3. The R code used to produce these figures can be found in Appendix A. >From the Item Response Category Characteristic Curves we observe that there is low probability of endorsing the first option, "very concerned", for relatively high latent trait levels. This indicates that the questions asked are not considered as major environmental issues by the subjects interviewed. The same conclusion is also reached by the Test Information Curve from which we can observe that the set of six questions provides 89% of the total information for high latent trait levels. Finally, the Item Information Curves indicate that items LeadPetrol and Nuclear provide little information in the whole latent trait continuum. In order to check this numerically the `information()` function is used:

```
R> information(fit2, c(-4, 4))
```

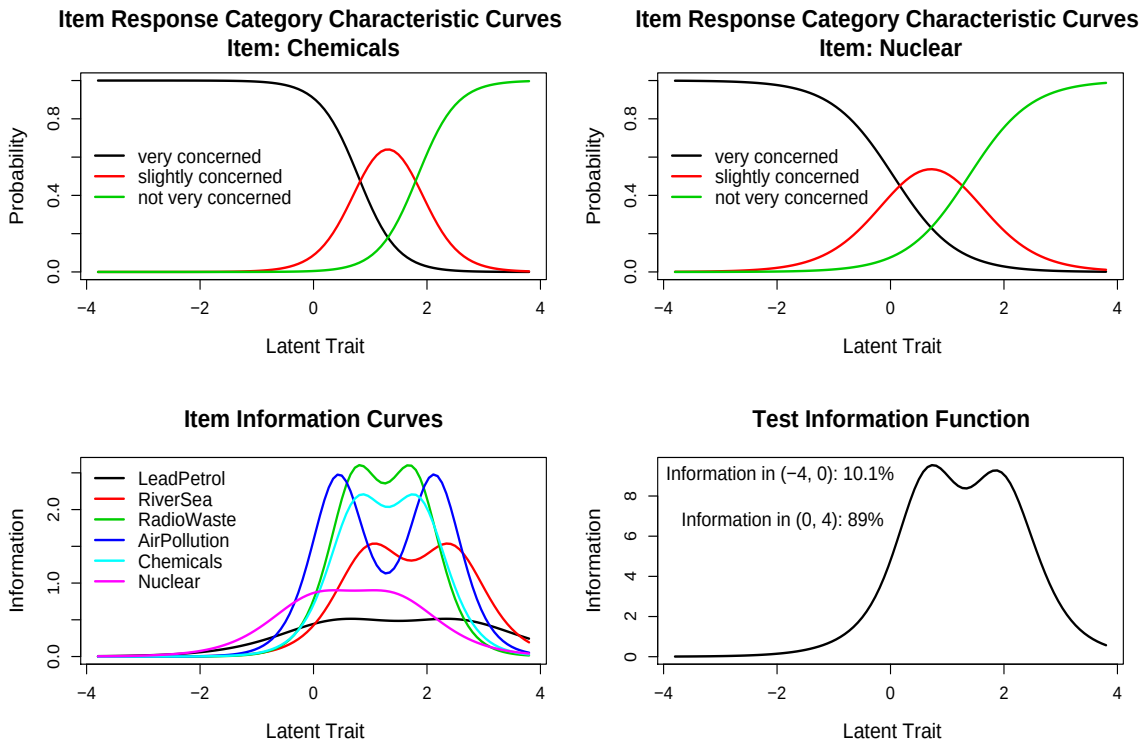

Figure 3: Item Characteristic for the last 2 items, Item Information and Test Information Curves for the Environment data-set under the unconstrained GRM model.

```
Call:
grm(data = Environment)

Total Information = 26.91
Information in (-4, 4) = 26.66 (99.08%)
Based on all the items

R> information(fit2, c(-4, 4), items = c(1, 6))

Call:
grm(data = Environment)

Total Information = 5.48
Information in (-4, 4) = 5.3 (96.72%)
Based on items 1, 6
```

We observe that these two items provide only the 20.36% (i.e., $100 \times 5.48/26.91$) of the total information, and thus they could probably be excluded from a similar future study. Finally, a useful comparison between items can be achieved by plotting the ICCs for each category separately. For the Environment data this comparison is depicted in Figure 4; the required commands can be found in Appendix A. An interesting feature of Figure 4

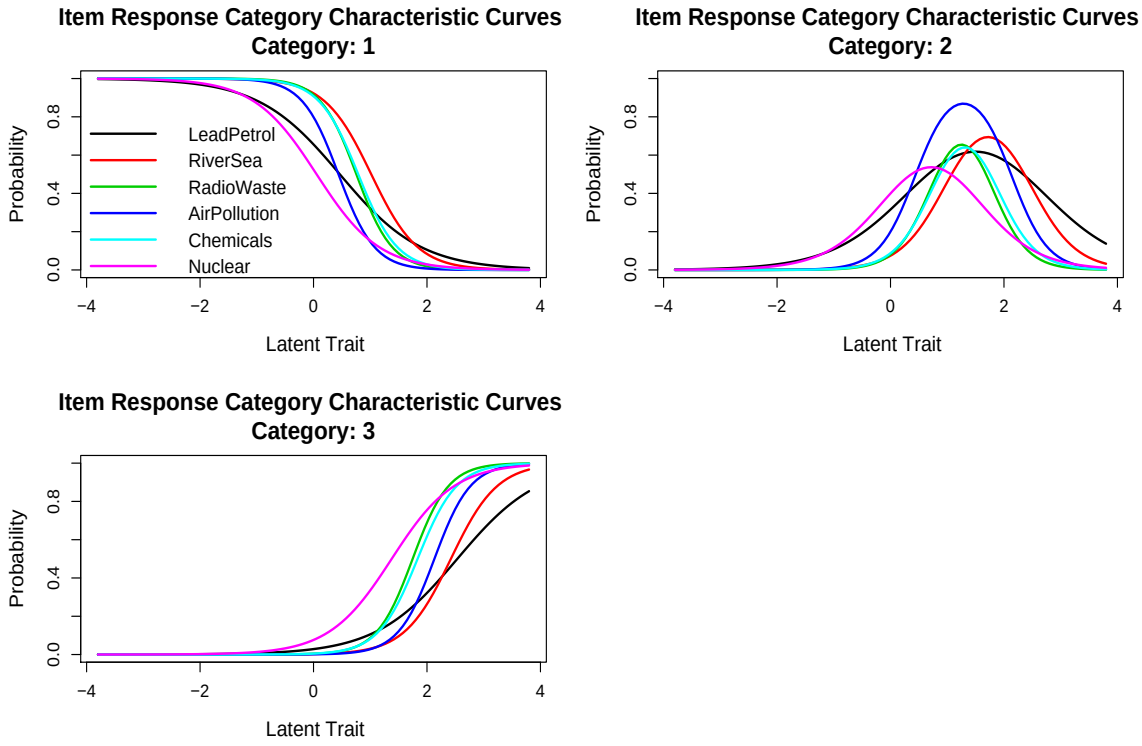

Figure 4: Item Characteristic Curves for each category separately for the Environment dataset under the unconstrained GRM model.

is that items RadioWaste and Chemicals have nearly identical characteristic curves for all categories, indicating that these two items are probably regarded to have the same effect on the environment.

4. Extra features of ltm and future development plans

The R package **ltm** provides a flexible framework for basic IRT analyses that covers some of the most common models for dichotomous and polytomous data. The main functions of the package have already been presented, but there are some additional features that we discuss here. These features mainly concern the function `ltm()`; in particular, `ltm()` fits latent trait models with one or two latent variables, allowing also for the incorporation of nonlinear terms between them as discussed in Rizopoulos and Moustaki (2006). This latter feature might prove useful in situations in which correlation between the latent variables is plausible. An example, that we briefly present here and for which such relationships provide reasonable interpretations, is the data taken from a section of the 1990 Workplace Industrial Relation Survey (WIRS) that deals with management/worker consultation in firms. The object `WIRS` in **ltm** provides a subset of these data that consists of 1005 firms and concerns non-manual workers. The aim of the survey was to investigate two types of consultation, namely *formal* and *informal*, and thus the use of two latent variables seems well justified. However, the fit of the two-factor model to the three-way margins is not successful, with some triplets of items having high residual values. Here we extend the simple two-factor model and allow for an

interaction between the two latent variables. The two-factor and the interaction models are fitted by `ltm()` using the following commands:

```
R> fit1 <- ltm(WIRS ~ z1 + z2)
R> fit2 <- ltm(WIRS ~ z1 * z2)
R> anova(fit1, fit2)

 Likelihood Ratio Table
         AIC     BIC  log.Lik   LRT df p.value
fit1 6719.08 6807.51 -3341.54
fit2 6634.42 6752.33 -3293.21 96.66  6  <0.001
```

The significant p-value suggests that this extension provides a better fit to the data than the two-factor model. This is also supported by the fit to the three-way margins in which all residuals have acceptable values. However, the inclusion of the interaction term not only improves the fit but also has an intuitive interpretation. In particular, the parameter estimates under the interaction model are

```
R> fit2

Call:
ltm(formula = WIRS ~ z1 * z2)

Coefficients:
        (Intercept)     z1      z2   z1:z2
Item 1       -1.097  0.271  -3.317  -1.160
Item 2        0.944  1.091   2.173   2.955
Item 3       -1.458  1.811  -0.314   0.543
Item 4       -1.465  1.116   0.556   0.269
Item 5       -1.083  2.133  -0.466   1.147
Item 6       -2.753  1.698  -0.935   1.072

Log.Lik: -3293.212
```

First, we should note that these estimates are under the additive parameterization (1). Second, if we change the signs for both the second factor `z2` and the interaction term `z1:z2`, which is the same solution but rotated, we observe that the first factor has high factor loadings for items three to six, which correspond to formal types of consultation, whereas the second factor has high loading for the first item which corresponds to the mains type of informal consultation. Item two has relatively high loadings for both factors implying that this item is probably regarded as a general type of consultation. The interaction term estimates have, for the majority of the items, a negative sign indicating that the more a firm uses one type of consultation, the smaller the probability of using the other type is. Finally, we should mention that latent trait models with nonlinear terms may lead to likelihood surfaces with local maxima. Thus, it is is advisable to investigate sensitivity to starting values using the `"random"` option in the `start.val` argument of `ltm()`.

A final feature that we will discuss here concerns the estimation of factor scores. Factor scores are usually calculated as the mode of the posterior distribution for each sample unit

$$\hat{z}_m = \arg\max_z \{p(x_m|z_m;\theta)p(z_m)\}. \tag{6}$$

Function `factor.scores()` calculates these modes using `optim()`. Note that in (6) we typically replace the true parameter values θ by their maximum likelihood estimate $\hat{\theta}$. Thus, in small samples we ignore the variability of plugging-in estimates instead of the true parameter values. To take this into account, the `factor.scores()` function offers the option to compute factor scores using a multiple imputation like approach (i.e., specified by the `method` argument), in which the uncertainty about the true parameter values is explicitly acknowledged; more details can be found in Rizopoulos and Moustaki (2006). Moreover, `factor.scores()` provides also the option, for `ltm` objects, to compute Component Scores as described in Bartholomew *et al.* (2002, Section 7.5).

Package **ltm** is still under active development. Future plans include extra functions to fit IRT models for ordinal (i.e., the partial credit and the rating scale models) and nominal manifest variables.

Acknowledgments

This work was partially supported by U.S. Army Medical Research and Materiel Command under Contract No. W81XWH-06-P-0124. The views, opinions and findings contained in this report are those of the author and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

The author thanks the editor, the associate editor, and the two referees for their constructive and thoughtful comments that substantially improved the article.

References

Agresti A (2002). *Categorical Data Analysis*. Wiley, New Jersey, 2nd edition.

Baker F, Kim SH (2004). *Item Response Theory*. Marcel Dekker, New York, 2nd edition.

Bartholomew D, Knott M (1999). *Latent Variable Models and Factor Analysis*. Arnold, London, 2nd edition.

Bartholomew D, Steele F, Moustaki I, Galbraith J (2002). *The Analysis and Interpretation of Multivariate Data for Social Scientists*. Chapman & Hall, London.

Birnbaum A (1968). "Some Latent Trait Models and Their Use in Inferring an Examinee's Ability." In F Lord, M Novick (eds.), "Statistical Theories of Mental Test Scores," Addison-Wesley, Reading, MA.

Bock R, Lieberman M (1970). "Fitting a Response Model for n Dichotomously Scored Items." *Psychometrika*, **35**, 179–197.

Brook L, Taylor B, Prior G (1991). *British Social Attitudes, 1990, Survey*. SCPR, London.

Mair P, Hatzinger R (2006). **eRm***: Estimating Extended Rasch Models.* R package version 0.3.

Martin A, Quinn K (2006). **MCMCpack***: Markov Chain Monte Carlo (MCMC) Package*. R package version 0.7-3, URL http://mcmcpack.wustl.edu/.

Pinheiro J, Bates D (1995). "Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model." *Journal of Computational and Graphical Statistics*, **4**, 12–35.

Rasch G (1960). *Probabilistic Models for some Intelligence and Attainment Tests*. Paedagogike Institut, Copenhagen.

R Development Core Team (2006). *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/.

Rizopoulos D, Moustaki I (2006). "Generalized Latent Variable Models with Non-Linear Effects." Submitted for publication.

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

Skrondal A, Rabe-Hesketh S (2004). *Generalized Latent Variable Modeling: Multilevel, Longitudinal and Structural Equation Models*. Chapman & Hall, Boca Raton, FL.

van der Linden W, Hambleton R (1997). *Handbook of Modern Item Response Theory*. Springer-Verlag, New York.

Zickar M (2002). "Modeling Data with Polytomous Item Response Theory." In F Drasgow, N Schmitt (eds.), "Measuring and Analyzing Behavior in Organizations," Jossey-Bass, San Francisco.

A. R code

All R code used in this paper is also available in a source code file `v17i05.R` together with the paper. Additionally, the R commands used to produce the figures are provided in the following. Figure 1 can be obtained via:

```
R> par(mfrow = c(2, 2))

R> plot(fit2, legend = TRUE, cx = "bottomright", lwd = 3,
+       cex.main = 1.5, cex.lab = 1.3, cex = 1.1)

R> plot(fit2, type = "IIC", annot = FALSE, lwd = 3, cex.main = 1.5,
+       cex.lab = 1.3)

R> plot(fit2, type = "IIC", items = 0, lwd = 3, cex.main = 1.5,
+       cex.lab = 1.3)

R> plot(0:1, 0:1, type = "n", ann = FALSE, axes = FALSE)
R> info1 <- information(fit2, c(-4, 0))
R> info2 <- information(fit2, c(0, 4))
R> text(0.5, 0.5, labels = paste("Total Information:", round(info1$InfoTotal, 3),
+       "\n\nInformation in (-4, 0):", round(info1$InfoRange, 3),
+       paste("(", round(100 * info1$PropRange, 2), "%)", sep = ""),
+       "\n\nInformation in (0, 4):", round(info2$InfoRange, 3),
+       paste("(", round(100 * info2$PropRange, 2), "%)", sep = "")), cex = 1.5)
```

The R commands used to produce Figures 2 and 3 are the following:

```
R> par(mfrow = c(2, 2))

R> plot(fit2, lwd = 2, cex = 1.2, legend = TRUE, cx = "left",
+       xlab = "Latent Trait", cex.main = 1.5, cex.lab = 1.3, cex.axis = 1.1)

R> plot(fit2, type = "IIC", lwd = 2, cex = 1.2, legend = TRUE, cx = "topleft",
+       xlab = "Latent Trait", cex.main = 1.5, cex.lab = 1.3, cex.axis = 1.1)

R> plot(fit2, type = "IIC", items = 0, lwd = 2, xlab = "Latent Trait",
+       cex.main = 1.5, cex.lab = 1.3, cex.axis = 1.1)

R> info1 <- information(fit2, c(-4, 0))
R> info2 <- information(fit2, c(0, 4))
R> text(-1.9, 8, labels = paste("Information in (-4, 0):",
+       paste(round(100 * info1$PropRange, 1), "%", sep = ""),
+       "\n\nInformation in (0, 4):",
+       paste(round(100 * info2$PropRange, 1), "%", sep = "")), cex = 1.2)
```

The R commands used to produce Figure 4 are the following:

```
R> par(mfrow = c(2, 2))

R> plot(fit2, category = 1, lwd = 2, cex = 1.2, legend = TRUE, cx = -4.5,
+       cy = 0.85, xlab = "Latent Trait", cex.main = 1.5, cex.lab = 1.3,
+       cex.axis = 1.1)

R> for (ctg in 2:3) {
+      plot(fit2, category = ctg, lwd = 2, cex = 1.2, annot = FALSE,
+           xlab = "Latent Trait", cex.main = 1.5, cex.lab = 1.3,
+           cex.axis = 1.1)
+ }
```

Affiliation:

Dimitris Rizopoulos
Biostatistical Centre
Catholic University of Leuven
U.Z. St. Rafaël, Kapucijnenvoer 35
B-3000 Leuven, Belgium
E-mail: dimitris.rizopoulos@med.kuleuven.be
URL: http://www.student.kuleuven.be/~m0390867/dimitris.htm

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 17, Issue 5 *Submitted:* 2006-05-08
November 2006 *Accepted:* 2006-11-20