IRT Models with Relaxed Assumptions in eRm
(draft)

Reinhold Hatzinger and Thomas Rusch

Linear logistic models with relaxed assumptions (LLRA) can be thought of as generalized *Rasch* models with multidimensional latent trait parameters where change is modelled as a function of treatment (main) effects, treatment interactions, and trend effects. CML estimation allows for the separation of "structural" treatment effect parameters and "incidential" trait parameters. Consequently, results about the structural effect parameters are completely independent of the multivariate distribution of trait parameters in the sample of subjects (principle of specific objectivity). Relaxed assumptions mean that neither unidimensionality of the items nor distributional assumptions about the population of subjects are required, cf. Fischer (1989)). In this paper we describe how different LLRA models can be fitted in **R** (R Development Core Team, 2008) using the **eRm** package (Mair and Hatzinger, 2008). Among these we discuss in Chapter 1 LLRAs for dichotomous responses with time effects (Sec. 1.1) or time and treatment effects (Sec. 1.2) for two time points and more than two time points (Sec. 1.3). In Section 1.4 we show how LLRAs can be used to estimate treatment and time effects over unidimensional subscales. Chapter 2 describes LLRAs for polytomous responses, including the Partial Credit Model approach (Sec. 2.1) for equal and different number of categories and the Rating Scale Model approach (Sec. 2.2) for two or more time points each. How to generalise effects over unidimensional subscales in the polytomous case is described in Section 2.4. Additionally, an appendix briefly explaining the useful concept of Kronecker products is included.

1. Dichotomous Responses

The most simple LLRA is a model for dichotomous responses at two time points, T_1 and T_2.

$$(1) \qquad P(X_{vi1} = 1|T_1) = \frac{\exp(\theta_{vi})}{1 + \exp(\theta_{vi})},$$

$$(2) \qquad P(X_{vi2} = 1|T_2) = \frac{\exp(\theta'_{vi})}{1 + \exp(\theta'_{vi})} = \frac{\exp(\theta_{vi} + \delta_{vi})}{1 + \exp(\theta_{vi} + \delta_{vi})},$$

where θ_{vi} is the location of subject v on the i-th latent trait at T_1, and $\delta_{vi} = \theta'_{vi} - \theta_{vi}$ is the amount of change of subject v on this particular trait i. There may be as many latent traits as items (multidimensionality) or items considered to measure the same latent trait may be grouped together. The flexibility of these models arises from the possibility to reparameterise δ_{vi} as

$$(3) \qquad \delta_{vi} = \mathbf{w}_i^T \boldsymbol{\eta}$$

where $\mathbf{w}$ is a vector of covariate values for item i, $i = 1, \ldots, k$ and $\boldsymbol{\eta}$ is a vector of parameters, typically describing treatments, interactions between treatments, and trends.

When fitting such a model, a different data structure has to be used compared to unidimensional models assuming homogeneous items. The usual data structure is displayed in Table 1.

TABLE 1. Data structure for models with repeated measurements.

Real Persons	T_1				T_2			
S_1	x_{111}	x_{121}	$\dots$	x_{1k1}	x_{112}	x_{122}	$\dots$	x_{1k2}
S_2	x_{211}	x_{221}	$\dots$	x_{2k1}	x_{212}	x_{222}	$\dots$	x_{1k2}
$\vdots$		$\vdots$				$\vdots$		
S_n	x_{n11}	x_{n21}	$\dots$	x_{nk1}	x_{n12}	x_{n22}	$\dots$	x_{nk2}

In models with relaxed assumptions however, the data have to be rearranged such that each row consists of the responses to a particular item for all time points, as is portrayed in Table 2.

TABLE 2. Modified data structure for models with relaxed assumptions.

Virtual Persons	T_1	T_2
S^*_{11}	x_{111}	x_{112}
S^*_{21}	x_{211}	x_{212}
$\vdots$	$\vdots$	$\vdots$
S^*_{n1}	x_{n11}	x_{n12}
S^*_{12}	x_{121}	x_{122}
S^*_{22}	x_{221}	x_{222}
$\vdots$	$\vdots$	$\vdots$
S^*_{n2}	x_{n21}	x_{n22}
$\vdots$	$\vdots$	$\vdots$
S^*_{1k}	x_{1k1}	x_{1k2}
S^*_{2k}	x_{2k1}	x_{2k2}
$\vdots$	$\vdots$	$\vdots$
S^*_{nk}	x_{nk1}	x_{nk2}

In fact, this modified data matrix is a simplified version of the real data structure in LLRAs (which includes structurally missing observations for certain virtual items (cf. Fischer (1995, p 163))

However, for ease of presentation and since this structure corresponds to the data structure used for input into *Rasch* model software (e.g., into **eRm**) we omit the more complicated representation.

The particular form of the design matrix depends on the study design and hypotheses. We can mainly distinguish two types of design matrices. One type, where we just look for changes over time and the other type, where additional variables describing subjects are taken into account (these may be experimental conditions and/or observed characteristics such as gender).

1.1. **Time effects**

Restricting ourselves to two time points, the general form of the first type of design matrices emerges when we specify k different trend parameters τ for the k items.

TABLE 3. Design matrix for trend parameters for each item.

	τ_1	τ_2	$\cdots$	τ_k
Item 1 – T_1				
Item 2 – T_1				
$\vdots$				
Item k – T_1				
Item 1 – T_2	1			
Item 2 – T_2		1		
$\vdots$			$\ddots$	
Item k – T_2				1

When using the **eRm** package we need a further design structure - the *item assignment*. Since different real items have been answered by the same subject but are treated as being the same item answered by different virtual subjects (cf. Tables 1 and 2) we need an additional specifier to identify which virtual subject has responded to which real item. Recalling the modified data structure the item assignment vector is

Virtual Persons	T_1	T_2
S^*_{11}	x_{111}	x_{112}
S^*_{21}	x_{211}	x_{212}
$\vdots$	$\vdots$	$\vdots$
S^*_{n1}	x_{n11}	x_{n12}
S^*_{12}	x_{121}	x_{122}
S^*_{22}	x_{221}	x_{222}
$\vdots$	$\vdots$	$\vdots$
S^*_{n2}	x_{n21}	x_{n22}
$\vdots$	$\vdots$	$\vdots$
S^*_{1k}	x_{1k1}	x_{1k2}
S^*_{2k}	x_{2k1}	x_{2k2}
$\vdots$	$\vdots$	$\vdots$
S^*_{nk}	x_{nk1}	x_{nk2}

Item Assignment
1
1
$\vdots$
1
2
2
$\vdots$
2
$\vdots$
k
k
$\vdots$
k

Example 1: 100 subjects have responded to 3 items at 2 occasions. We want to investigate if there is a change between T_1 and T_2.

First the **eRm** package is loaded.

```
> library(eRm)
```

The data are given in the usual form in file `llra_ex1.dat` (each row corresponds to one person, cf. Table 1).

```
> dat <- read.table("llra_ex1.dat", header = TRUE)
```

The first few lines of the data are

```
> head(dat)
```

```
  T1I1 T1I2 T1I3 T2I1 T2I2 T2I3
1    0    1    1    1    0    1
2    1    1    0    1    0    1
3    1    0    0    1    0    1
4    1    1    0    1    0    1
5    1    0    1    1    1    1
6    1    1    0    0    0    1
```

To obtain the data structure that can be used for an LLRA we have to rearrange the data set to fit the form of Table 2. For example, we could use a command such as

```
> data <- matrix(unlist(dat), nc = 2)
```

The first few lines of the modified data matrix in `data` are

```
> head(data)
```

```
     [,1] [,2]
[1,]    0    1
[2,]    1    1
[3,]    1    1
[4,]    1    1
[5,]    1    1
[6,]    1    0
```

Since R stores matrices columnwise, the ordering in `data` produced by the above `matrix()` command is:

row	subject	number of (real) item at T_1	number of (real) item at T_2
`[1]`	1	1	1
`[2]`:	2	1	1
⋮	⋮	⋮	⋮
`[n]`:	n	1	1
`[n+1]`:	1	2	2
⋮	⋮	⋮	⋮
`[2n]`:	n	2	2
`[2n+1]`:	1	3	3
⋮	⋮	⋮	⋮
`[(k-1)n+v]`:	v	k	k
⋮	⋮	⋮	⋮
`[(k-1)n+n]`:	n	k	k

Accordingly, the first 100 rows in `data` correspond to columns 1 and 4 of `dat`, the two columns which consist of the responses to (real) item 1 at times T_1 and T_2. The next 100 rows correspond to columns 2 and 5 in `dat`, etc. To know which (real) item is in which row of the modified data matrix is crucial for setting up the item assignment vector properly.

In fact, the ordering of the rows of the modified data matrix does not matter as long as the item assignment vector is correctly specified.

We now define a model where we want to estimate 3 change parameters τ (cf. Table 3), i.e., it is assumed that the amount of change is different for all items. This model can be estimated as an LPCM. To assign the (real) items to the appropriate rows of the design matrix we need the items assignment vector as discussed above

```
> Igrps <- as.numeric(gl(3, 100, 300))
```

and the design matrix as given in Table 3. An easy way to construct this design matrix is by using a Kronecker product (for some basics on Kronecker products see Appendix A)

```
> design.3 <- c(0, 1) %x% diag(3)
> colnames(design.3) <- paste("TAU", 1:3, sep = "")
```

The model is now fitted using the command

```
> res.llra.3 <- LPCM(data, W = design.3, mpoints = 2, groupvec = Igrps,
+     sum0 = FALSE)
```

The result is

```
> summary(res.llra.3)

Results of LPCM estimation:

Call:  LPCM(X = data, W = design.3, mpoints = 2, groupvec = Igrps, sum0 = FALSE)

Conditional log-likelihood: -104.3815
Number of iterations: 5
Number of parameters: 3

Basic Parameters (eta) with 0.95 CI:
     Estimate Std. Error lower CI upper CI
TAU1    1.050      0.311    0.441    1.658
TAU2   -0.452      0.279   -0.999    0.095
TAU3   -0.074      0.272   -0.608    0.460

Item Easiness Parameters (beta) with 0.95 CI:
            Estimate Std. Error lower CI upper CI
I1.c1 t1 g1    0.000      0.000    0.000    0.000
I1.c1 t1 g2    0.000      0.000    0.000    0.000
I1.c1 t1 g3    0.000      0.000    0.000    0.000
I1.c1 t2 g1    1.050      0.311    0.441    1.658
I1.c1 t2 g2   -0.452      0.279   -0.999    0.095
I1.c1 t2 g3   -0.074      0.272   -0.608    0.460
```

For models with two time points, the `Basic Parameters (eta)` give the log odds for the response pattern ratio (0,1)/(1,0) for each item group. The frequency tables for the three items (`INDICES`) at T_1 (`V1`) and T_2 (`V2`) are

```
> ftab <- by(data, Igrps, table)
> ftab

INDICES: 1
   V2
```

```
V1   0  1
  0 10 40
  1 14 36
-----------------------------------------------------
INDICES: 2
   V2
V1   0  1
  0 31 21
  1 33 15
-----------------------------------------------------
INDICES: 3
   V2
V1   0  1
  0 25 26
  1 28 21
```

For item 1, e.g., 10 subjects responded with 0 at both time points whereas 40 changed from response 0 to 1. The log odds for the response pattern ratio (0,1)/(1,0) for item 1 is

```
> cat("log(", ftab[[1]][1, 2], "/", ftab[[1]][2, 1], ") = ",
+     log(ftab[[1]][1, 2]/ftab[[1]][2, 1]), "\n")

log( 40 / 14 ) =  1.049822
```

which corresponds to the value given for the estimate `TAU1`.

To continue the analysis, we alternatively may specify a model with just one change parameter for the items, i.e., the trend is supposed to generalise over all items. A different design matrix is needed. In general, if trend effects are hypothesised to be the same for some items the corresponding columns in the design matrix have to be collapsed

```
> design.0 <- as.matrix(rowSums(design.3))
> colnames(design.0) <- "TAU1"
> design.0

     TAU1
[1,]    0
[2,]    0
[3,]    0
[4,]    1
[5,]    1
[6,]    1
```

The model where the trend is supposed to generalise over all items is

```
> res.llra.0 <- LPCM(data, W = design.0, mpoints = 2, groupvec = Igrps,
+     sum0 = F)
> print(res.llra.0)

Results of LPCM estimation:

Call:  LPCM(X = data, W = design.0, mpoints = 2, groupvec = Igrps, sum0 = F)

Conditional log-likelihood: -111.845
Number of iterations: 3
Number of parameters: 1
```

```
Basic Parameters eta:
             TAU1
Estimate 0.1484195
Std.Err  0.1575683
```

To compare these two models we can perform a likelihood ratio test. Since we will need such calculations again, we define a little function `lrtst()`.

```
> lrtst <- function(model.1, model.2) {
+     lrstat <- 2 * abs(model.1$loglik - model.2$loglik)
+     df <- abs(model.1$npar - model.2$npar)
+     prb <- round(1 - pchisq(lrstat, df), digits = 3)
+     cat("Likelihood ratio statistic:", lrstat, "df =",
+         df, "p =", prb, "\n")
+ }
```

Applying it to our two models yields

```
> lrtst(res.llra.0, res.llra.3)
```

```
Likelihood ratio statistic: 14.92703 df = 2 p = 0.001
```

The result indicates that the trend is not the same for all items. However, examining the summary for the output object `res.llra.3` we can see that the confidence intervals for `TAU2` and `TAU3` overlap. We could therefore test for $\tau_2 = \tau_3$. This is simply achieved by again redefining the design matrix (i.e., by collapsing columns 2 and 3 of `design.3`) and refitting the model

```
> design.2 <- cbind(design.3[, 1], rowSums(design.3[, 2:3]))
> colnames(design.2) <- c("TAU1", "TAU23")
> design.2
```

```
     TAU1 TAU23
[1,]    0     0
[2,]    0     0
[3,]    0     0
[4,]    1     0
[5,]    0     1
[6,]    0     1
```

```
> res.llra.2 <- LPCM(data, W = design.2, mpoints = 2, groupvec = Igrps,
+     sum0 = FALSE)
> res.llra.2
```

```
Results of LPCM estimation:

Call:  LPCM(X = data, W = design.2, mpoints = 2, groupvec = Igrps, sum0 = FALSE)

Conditional log-likelihood: -104.8531
Number of iterations: 5
Number of parameters: 2

Basic Parameters eta:
             TAU1      TAU23
```

```
Estimate 1.0498235 -0.2607262
Std.Err  0.3105375  0.1940865
```

```
> lrtst(res.llra.2, res.llra.3)
```

```
Likelihood ratio statistic: 0.943219 df = 1 p = 0.331
```

The LR-test between the model with separate trend parameters for each item (`res.llra.3`) and the model with a common trend parameter for items 2 and 3 (`res.llra.2`) is not significant. We can thus conclude that the amount of change is the same for item 2 and item 3 but different for item 1. However, there seems to be no change at all for items 2 and 3. We can test this hypothesis by omitting columns 2 and 3 from the design matrix.

```
> design.1 <- as.matrix(design.3[, 1])
> res.llra.1 <- LPCM(data, W = design.1, mpoints = 2, groupvec = Igrps,
+     sum0 = FALSE)
> res.llra.1
```

```
Results of LPCM estimation:

Call:  LPCM(X = data, W = design.1, mpoints = 2, groupvec = Igrps, sum0 = FALSE)

Conditional log-likelihood: -105.7631
Number of iterations: 5
Number of parameters: 1

Basic Parameters eta:
             eta 1
Estimate 1.0498216
Std.Err  0.3105373
```

```
> lrtst(res.llra.1, res.llra.2)
```

```
Likelihood ratio statistic: 1.819932 df = 1 p = 0.177
```

The LR-test supports the hypothesis of no change for items 2 and 3.

1.2. **Time and treatment effects**

We now want to estimate the effect of different treatments or the effect of certain covariates over time additionally to the (general) time effects (trends). The introduction of such treatment effects is straightforward. Suppose we have a treatment and a control group (or a group of men and women) and we want to estimate different treatment (or gender) effects λ_i and different trends τ_i for each item i. The design matrix where the rows are ordered according to the requirements of the **eRm** package is given in Table 4.

Other specific hypotheses (like generalisation of treatment or trend effects over several items) are obtained by collapsing appropriate columns in analogy to Example 1.

General remark concerning the construction of the rows of the design matrix: The slowest index are time points. Within time points are items, and within items are (treatment) groups. The fastest index are response categories in case of polytomous items.

Time Points $\sqsubset$ Items $\sqsubset$ (Treatment) Groups $\sqsubset$ Categories

TABLE 4. Design matrix for different treatment and trend parameters for each item for a treatment group (TG) and a control group (CG).

		λ_1	λ_2	$\cdots$	λ_k	τ_1	τ_2	$\cdots$	τ_k
T_1	Item 1 – TG								
	Item 1 – CG								
	Item 2 – TG								
	Item 2 – CG								
	$\vdots$								
	Item k – TG								
	Item k – CG								
T_2	Item 1 – TG	1				1			
	Item 1 – CG					1			
	Item 2 – TG		1				1		
	Item 2 – CG						1		
	$\vdots$			$\vdots$				$\vdots$	
	Item k – TG				1				1
	Item k – CG								1

Example 2 (Example 1 continued)**:** Suppose the first 50 subjects received a treatment whereas the second 50 did not (control group). We want to additionally estimate the effect of the treatment on each item separately.

We first have to set up the design matrix (cf. Table 4). This can easily be done by generating a zero matrix with appropriate dimensions

```
> design4 <- (matrix(0, nrow = 12, ncol = 6))
```

and then use the data editor and insert 1s in the appropriate positions.

```
> fix(design4)
```

Alternatively, we could use the design matrix `design.3` from above (cf Table 3) and apply Kronecker products[1].

```
> design4 <- cbind(design.3 %x% c(1, 0), design.3 %x% c(1,
+     1))
> design4
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
 [1,]    0    0    0    0    0    0
 [2,]    0    0    0    0    0    0
 [3,]    0    0    0    0    0    0
 [4,]    0    0    0    0    0    0
 [5,]    0    0    0    0    0    0
```

[1]The Kronecker product (`%x%`) of the matrix `design.3` with the (column) vector `c(1,0)` doubles each row of `design.3` (since the length of `c(1,0)` is 2) where in each case the first resulting row is multiplied by 1 and the second multiplied by 0.

```
 [6,]    0    0    0    0    0    0
 [7,]    1    0    0    1    0    0
 [8,]    0    0    0    1    0    0
 [9,]    0    1    0    0    1    0
[10,]    0    0    0    0    1    0
[11,]    0    0    1    0    0    1
[12,]    0    0    0    0    0    1
```

We again need an item assignment vector, but now we have to take account of the treatment groups, i.e., we have to identify the subjects in a certain treatment group who responded to a certain item. This new assignment is therefore a combination of treatment group × items. In general, a separate group has to be defined for each such (treatment group × item) – combination. Please note, that the assignment vector determines which reponses belong together, i.e., it has the same length as the (modified) data matrix and must be specified such that each row is assigned to the corresponding (treatment group × item) – combination. Table 5 gives an illustration.

TABLE 5. Data structure for designs with treatments

		Virtual Persons	T_1	T_2	Assignment Group
Item 1	TG	$S^*_{(TG)11}$	$x_{(TG)111}$	$x_{(TG)112}$	1
		$\vdots$	$\vdots$	$\vdots$	$\vdots$
		$S^*_{(TG)n1}$	$x_{(TG)n11}$	$x_{(TG)n12}$	1
	CG	$S^*_{(CG)11}$	$x_{(CG)111}$	$x_{(CG)112}$	2
		$\vdots$	$\vdots$	$\vdots$	$\vdots$
		$S^*_{(CG)n1}$	$x_{(CG)n11}$	$x_{(CG)n12}$	2
Item 2	TG	$S^*_{(TG)12}$	$x_{(TG)121}$	$x_{(TG)122}$	3
		$\vdots$	$\vdots$	$\vdots$	$\vdots$
		$S^*_{(TG)n2}$	$x_{(TG)n21}$	$x_{(TG)n22}$	3
	CG	$S^*_{(CG)12}$	$x_{(CG)121}$	$x_{(CG)122}$	4
		$\vdots$	$\vdots$	$\vdots$	$\vdots$
		$S^*_{(CG)n2}$	$x_{(CG)211}$	$x_{(CG)n12}$	4
		$\vdots$	$\vdots$	$\vdots$	$\vdots$
Item k	TG	$S^*_{(TG)1k}$	$x_{(TG)1k1}$	$x_{(TG)1k2}$	$2k-1$
		$\vdots$	$\vdots$	$\vdots$	$\vdots$
		$S^*_{(TG)nk}$	$x_{(TG)nk1}$	$x_{(TG)nk2}$	$2k-1$
	CG	$S^*_{(CG)1k}$	$x_{(CG)1k1}$	$x_{(CG)1k2}$	$2k$
		$\vdots$	$\vdots$	$\vdots$	$\vdots$
		$S^*_{(CG)nk}$	$x_{(CG)nk1}$	$x_{(CG)nk2}$	$2k$

In our example we need 6 assignment groups (treatment/control and 3 items) each of size 50. Having defined the assignment vector we fit the model with treatment and trend parameters for each item.

```
> grps6 <- as.numeric(gl(6, 50, 300))
> res.llra.4 <- LPCM(data, W = design4, mpoints = 2, groupvec = grps6,
```

```
+     sum0 = F)
> summary(res.llra.4)

Results of LPCM estimation:

Call:  LPCM(X = data, W = design4, mpoints = 2, groupvec = grps6, sum0 = F)

Conditional log-likelihood: -94.49457
Number of iterations: 16
Number of parameters: 6

Basic Parameters (eta) with 0.95 CI:
      Estimate Std. Error lower CI upper CI
eta 1    1.892      0.827    0.271    3.512
eta 2   -0.791      0.571   -1.910    0.328
eta 3    1.910      0.609    0.716    3.103
eta 4    0.460      0.369   -0.263    1.182
eta 5   -0.074      0.385   -0.829    0.681
eta 6   -0.965      0.415   -1.779   -0.151

Item Easiness Parameters (beta) with 0.95 CI:
           Estimate Std. Error lower CI upper CI
I1.c1 t1 g1    0.000      0.000    0.000    0.000
I1.c1 t1 g2    0.000      0.000    0.000    0.000
I1.c1 t1 g3    0.000      0.000    0.000    0.000
I1.c1 t1 g4    0.000      0.000    0.000    0.000
I1.c1 t1 g5    0.000      0.000    0.000    0.000
I1.c1 t1 g6    0.000      0.000    0.000    0.000
I1.c1 t2 g1    2.351      0.740    0.901    3.802
I1.c1 t2 g2    0.460      0.369   -0.263    1.182
I1.c1 t2 g3   -0.865      0.421   -1.691   -0.039
I1.c1 t2 g4   -0.074      0.385   -0.829    0.681
I1.c1 t2 g5    0.944      0.445    0.071    1.818
I1.c1 t2 g6   -0.965      0.415   -1.779   -0.151
```

Care has to be taken when interpreting the `Basic Parameters (eta)` since LLRA models are not hierarchical, i.e., the treatment effects are always nested within the time effects. For models with two time points and treatment groups, the `Item Parameters` give the log odds for the response pattern ratio (0,1)/(1,0) for each (item × treatment)-group. As can be seen from the output above (see `Basic Parameters (eta)`), the treatment effect for the first item `eta 1` (1.8918) is the difference of the log odds (i.e., the log odds-ratio), T_2 vs. T_1, for the treatment group (2.3514) and the control group (0.4595) for item 1.

```
> cat(res.llra.4$betapar[7:8], "\n")

2.351374 0.4595322

> cat(res.llra.4$betapar[7] - res.llra.4$betapar[8], "\n")

1.891842
```

Further hypotheses, e.g., generalisability of treatment effects over several items etc., can again be specified by collapsing appropriate columns of the design matrix.

Example 3 (Example 1 continued)**:** We now suspect that the treatment effects are the same for items 1 and 3 (as suggested by the estimates from model `res.llra.4`), i.e., $\lambda_1 = \lambda_3$.

The design matrix is obtained by collapsing columns 1 and 3 of `design4`

```
> design4a <- cbind(design4[, 1] + design4[, 3], design4[,
+     c(2, 4:6)])
> design4a
```

```
      [,1] [,2] [,3] [,4] [,5]
 [1,]    0    0    0    0    0
 [2,]    0    0    0    0    0
 [3,]    0    0    0    0    0
 [4,]    0    0    0    0    0
 [5,]    0    0    0    0    0
 [6,]    0    0    0    0    0
 [7,]    1    0    1    0    0
 [8,]    0    0    1    0    0
 [9,]    0    1    0    1    0
[10,]    0    0    0    1    0
[11,]    1    0    0    0    1
[12,]    0    0    0    0    1
```

Fitting the model yields

```
> res.llra.4a <- LPCM(data, W = design4a, mpoints = 2,
+     groupvec = grps6, sum0 = F)
> res.llra.4a

Results of LPCM estimation:

Call:  LPCM(X = data, W = design4a, mpoints = 2, groupvec = grps6, sum0 = F)

Conditional log-likelihood: -94.49471
Number of iterations: 12
Number of parameters: 5

Basic Parameters eta:
             eta 1      eta 2     eta 3       eta 4      eta 5
Estimate 1.9033262 -0.7908937 0.4572550 -0.07410331 -0.9621958
Std.Err  0.4907199  0.5709421 0.3441605  0.38516357  0.3798782
```

and the LR-test

```
> lrtst(res.llra.4a, res.llra.4)

Likelihood ratio statistic: 0.0002967158 df = 1 p = 0.986
```

supports the simplification. As an exercise, the reader might try to further reduce the model, e.g., by omitting nonsignificant terms.

1.3. More than two time points

Basically, everything mentioned so far also applies to models for more than two measurement points. There are, however, more possible hypotheses and therefore more different ways to specify the design matrix. Moreover, the parameters are no longer interpretable as simple log odds or log odds ratios but should be (more generally) considered as reflecting the amount of change on the latent trait.

Example 4: To illustrate a model with more than two repeated observations we shall use data from 100 subjects who have responded to 2 items at 3 occasions. The data are again given in usual form (each row corresponds to one real person).

We read the data from file `llra_ex2.dat` and inspect the first few lines

```
> dat2 <- read.table("llra_ex2.dat", header = TRUE)
> head(dat2)
```

```
  T1I1 T1I2 T2I1 T2I2 T3I1 T3I2
1    0    0    1    1    1    1
2    1    0    1    0    0    0
3    0    0    1    0    0    1
4    0    1    1    1    0    1
5    0    0    1    1    0    0
6    0    0    1    1    0    1
```

For LLRA analysis we again have to convert the data.

```
> data2 <- matrix(unlist(dat2), nc = 3)
> dim(data2)
```

```
[1] 200   3
```

The dimension is 200×3, i.e., there are 200 virtual subjects having responded to 3 virtual items. The first 100 lines correspond to the first real item, the other 100 lines to the second.

We first consider the hypothesis of different time effects for the items, where different amounts of change occur between timepoints T_1 and T_2, and T_1 and T_3. According to the rule

$$\text{Time Points} \quad \sqsubset \quad \text{Items} \quad \sqsubset \quad \text{(Treatment) Groups} \quad \sqsubset \quad \text{Categories}$$

we set up the design matrix as given in Table 6 and the assignment vector (for the two items), e.g., using the commands

```
> idx <- cbind(c(3, 5, 4, 6), 1:4)
> dsgn4 <- matrix(0, 6, 4)
> dsgn4[idx] <- 1
> I2grps <- as.numeric(gl(2, 100))
```

and fit the model

```
> res2.llra.4 <- LPCM(data2, dsgn4, mpoints = 3, groupvec = I2grps)
> res2.llra.4
```

TABLE 6. Design matrix for different trend parameters for the items i between T_1 and T_2 ($\tau_i^{t_2-t_1}$), and between T_1 and T_3 ($\tau_i^{t_3-t_1}$).

		$\tau_1^{t_2-t_1}$	$\tau_1^{t_3-t_1}$	$\tau_2^{t_2-t_1}$	$\tau_2^{t_3-t_1}$
T_1	Item 1				
	Item 2				
T_2	Item 1	1			
	Item 2			1	
T_3	Item 1		1		
	Item 2				1

```
Results of LPCM estimation:

Call:  LPCM(X = data2, W = dsgn4, mpoints = 3, groupvec = I2grps)

Conditional log-likelihood: -97.5181
Number of iterations: 13
Number of parameters: 4

Basic Parameters eta:
             eta 1         eta 2     eta 3     eta 4
Estimate 2.5535969 -1.267079e-06 1.8807291 3.0667937
Std.Err  0.4029427  3.751091e-01 0.4237687 0.4525402
```

The estimates show that for item 1 there is a significant change between T_1 and T_2 (`eta 1`) but no change between T_1 and T_3 (`eta 2`). For item 2 the picture is different. Here, the trend between T_1 and T_2 (`eta 3`) seems to continue up to T_3. The frequencies and the proportions of the item responses using the original data reflect this pattern.

```
> tab <- apply(dat2, 2, table)
> tab

  T1I1 T1I2 T2I1 T2I2 T3I1 T3I2
0   77   87   19   55   77   27
1   23   13   81   45   23   73

> proportions <- tab[2, ]/colSums(tab)
> proportions[c(1, 3, 5)]

T1I1 T2I1 T3I1
0.23 0.81 0.23

> proportions[c(2, 4, 6)]

T1I2 T2I2 T3I2
0.13 0.45 0.73
```

We could therefore try to simplify the model by introducing a linear trend for item 2 on the latent trait. Instead of two design columns for item 2 the whole design matrix is then

```
> dsgn3 <- cbind(dsgn4[, 1:2], c(0, 0, 0, 1, 0, 2))
> dsgn3

     [,1] [,2] [,3]
[1,]    0    0    0
```

```
[2,]    0    0    0
[3,]    1    0    0
[4,]    0    0    1
[5,]    0    1    0
[6,]    0    0    2
```

Fitting this models and testing if the simplification is admissible yields

```
> res2.llra.3 <- LPCM(data2, dsgn3, mpoints = 3, groupvec = I2grps)
> res2.llra.3

Results of LPCM estimation:

Call:  LPCM(X = data2, W = dsgn3, mpoints = 3, groupvec = I2grps)

Conditional log-likelihood: -98.20308
Number of iterations: 12
Number of parameters: 3

Basic Parameters eta:
             eta 1         eta 2     eta 3
Estimate 2.5535957 -4.460496e-06 1.4787503
Std.Err  0.4029424  3.751083e-01 0.2125844

> lrtst(res2.llra.4, res2.llra.3)

Likelihood ratio statistic: 1.369966 df = 1 p = 0.242
```

We conclude that for item 1 the amount of change towards higher probability of responses in category 1 increases between T_1 and T_2, but at T_3 decreases back to the same level as at T_1. For item 2, however, we can observe a positive continuing trend over all three time points.

1.4. **Modelling change over unidimensional subscales**

Sometimes it might be the case that not every item measures a particular dimension d on its own, but that a test comprises of a number of subscales each measuring a latent trait. The subscales are formed by a number of unidimensional items, measuring a single latent trait. Therefore we have a number of subscales corresponding to different dimensions and within each dimension unidimensional measurement is possible.

This means we need a multidimensional model that allows for any number (up to the number of items of course) of unidimensional subscales. This model combines the idea of the LLTM (measuring change over time for unidimensional scales) and the multidimensionality of the LLRA.

Employing LLTMs seperately for every dimension results in trend and treatment effects that are specific to this particular dimension, but generalisation of these effects over subscales is not possible. With the combined approach presented in the following, such a generalisation becomes feasible.

Estimation of such a model is possible with **eRm** if we use the concept of virtual subjects and virtual items. We need every subject S_v represented by only one single parameter θ_v, which means that for each subject S_v her multidimensional trait vector $(\theta_{v1}, \ldots, \theta_{vd}, \ldots, \theta_{vD})$ has to be distributed among D virtual subjects S^*_{vd}. We therefore get nD virtual subjects S^*_{vd} with trait parameters θ_{vd}. Now the parameters δ_{vi} from

the previous sections do have a special form, which we will denote with β^*_{git}, namely they represent the parameter of the item i, β_i, and the change $\delta_{gt} = \mathbf{w}^T_{gt}\boldsymbol{\eta}$, with $g = 1, \ldots, G$ denoting the number of treatment groups and $t = 1, \ldots, T$ the number of time points. This allows us to model change for the treatment groups and time points seperately either as being homogeneous in every subscale or over all subscales. Please note that if every item measures one dimension alone, this becomes a LLRA as described earlier.

Let us assume w.l.o.g that within each time point, the items are ordered according to the dimensions they measure, e.g. the first $s1$ items measure dimension 1, the next $s2$ items measure dimension 2 and so forth. If we denote with X_{gdt} the matrix of responses of all people S_v within group g to the items measuring dimension d at time t, we have the following usual data structure of repeated measurements in Table 7(if we only look on dimensions and groups).

TABLE 7. Data structure for repeated measurements on group and dimsension level

	T_1				T_t		
	D_1	$\ldots$	D_D		D_1	$\ldots$	D_D
G_1	X_{111}	$\ldots$	X_{1D1}	$\ldots$	X_{11t}	$\ldots$	X_{1Dt}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
G_g	X_{g11}	$\ldots$	X_{gD1}	$\ldots$	X_{g1t}	$\ldots$	X_{gDt}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
G_G	X_{G11}	$\ldots$	X_{GD1}	$\ldots$	X_{G1t}	$\ldots$	X_{GDt}

This data structure is a more general representation of the data structure in Table 1. Now to apply the model described above, the data must be rearranged in such a way that for every row of the new matrix there corresponds one single (virtual) person parameter θ_{vd} to each row and one single item parameter β^*_{git} to each column. We therefore get a data matrix with people $\times$ dimensions rows and items $\times$ time points $\times$ groups columns with a high number of data missing by design. The resulting data matrix[2] is shown in 8 ("-" are structurally missing data).

Unlike in case of the LLRA, if we use this data structure, we do not need an assignment vector, since in each row we have one (virtual) person answering to every (virtual) item of a dimension at all time points. Therefore it is clear which item has been answered by which person.

The next step is to set up the design matrix needed for this model and appropriate for this data structure. For convenience, we dropped the earlier convention on how to construct the rows of the design matrix because for these models it is much easier to construct the data and design matrix the following way:

$$\text{(Treatment) Groups} \sqsubset \text{Dimensions} \sqsubset \text{Time Points} \sqsubset \text{Categories}$$

Note that these order of rows in the design matrix corresponds one-to-one to the columns of Table 8. The design matrix is therefore a combination of LLTM design matrices and trend and treatment effects like in LLRAs. Table 9 shows how the design matrix for a single treatment group looks. For every group such a matrix has to be constructed and they must all be stacked upon each other. The easiest example would be just one treatment and one

[2]Please note that this is only one way of rearranging the data set. However, we think it is most convenient to arange the data in this blockdiagonal form.

control group, its design matrix would be a concatenation of two matrices like Table 9 upon each other with substituting zeros for ones in the columns of $\lambda_d (d = 1, \ldots, D)$ in the lower half (the control group). Obviously, the order of groups in the design matrix must reflect the ordering of groups in the rearranged data structure.

For the parameters of β within each unidimensional subscale one parameter has to be set to 0, which means deleting the corresponding column. In Table 9 it is always the first item of each dimension, e.g. β_1 and $\beta_{(\sum_d (sd-1))}$. This is because within each subscale a LLTM is estimated.

Example 5: A treatment group ($n = 50$) and a control group ($n = 30$) have responded to 6 items at 3 time points. W.l.o.g, the first 3 items measure one and the same latent dimension, items 4 to 6 measure a second dimension. We want to estimate a trend effect and a treatment effect that generalises over all dimensions, assuming unidimensional measurement within each dimension.

The data are given in the file `hyblltm.dat`. First we must modify them to look like in Table 8:

```
> dats <- as.matrix(read.table("hyblltm.dat", header = TRUE))
> D1.TG <- dats[1:50, c(1:3, 7:9, 13:15)]
> D2.TG <- dats[1:50, c(4:6, 10:12, 16:18)]
> D1.CG <- dats[51:80, c(1:3, 7:9, 13:15)]
> D2.CG <- dats[51:80, c(4:6, 10:12, 16:18)]
> null.TG <- matrix(NA, nrow = 50, ncol = 9)
> null.CG <- matrix(NA, nrow = 30, ncol = 9)
> D1TG <- cbind(D1.TG, null.TG, null.TG, null.TG)
> D2TG <- cbind(null.TG, D2.TG, null.TG, null.TG)
> D1CG <- cbind(null.CG, null.CG, D1.CG, null.CG)
> D2CG <- cbind(null.CG, null.CG, null.CG, D2.CG)
> data <- rbind(D1TG, D2TG, D1CG, D2CG)
```

TABLE 8. Rearranged data structure for multidimensional LLTM

G_1						...	G_G					
D_1		...		D_D		...	D_1		...		D_D	
T_1	T_2	...	...	T_1	T_2	...	T_1	T_2	...	...	T_1	T_2
X_{111}	X_{112}	-	-	-	-	...	-	-	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮	⋮	⋮
-	-	X_{1d1}	X_{1d2}	-	-	...	-	-	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮	⋮	⋮
-	-	-	-	X_{1D1}	X_{1D2}	...	-	-	-	-	-	-
-	-	-	-	-	-	...	X_{G11}	X_{G12}	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮	⋮	⋮
-	-	-	-	-	-	...	-	-	X_{Gd1}	X_{Gd2}	-	-
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮	⋮	⋮
-	-	-	-	-	-	...	-	-	-	-	X_{GD1}	X_{GD2}

TABLE 9. Design matrix for multidimensional LLTMs for a single treatment group.

			β_2	…	…	β_{s1}	…	$\beta_{(\sum_d (sd-1)+1)}$	…	…	$\beta_{(\sum_d sd)}$	λ_1	…	λ_D	τ_1	…	τ_D
G_1	D_1	T1	0														
				1													
					⋱												
						1											
		T2	0									1			1		
				1								1			1		
					⋱							1			1		
						1						1			1		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	D_D	T1						0									
									1								
										⋱							
											1						
		T2						0						1			1
									1					1			1
										⋱				1			1
											1			1			1

The design matrix can either be specified using a null matrix (filled with 0s) with appropriate dimension and `fix()` to enter the corresponding numbers, or to piece it together from some submatrices. We will illustrate the latter method since it is easier for larger design matrices. We start defining a design matrix for the LLTM in each dimension and at each time point, which is for 3 items

```
> temp1a <- diag(3)
> temp1a <- temp1a[, -1]
```

Next we must assign the LLTM design to the correct dimension, e.g. via

```
> temp1b <- matrix(0, nrow = 3, ncol = 2)
> temp1 <- cbind(temp1a, temp1b)
> temp2 <- cbind(temp1b, temp1a)
```

Since we have three time points we must set up this two dimensional LLTM three times

```
> tp <- rep(1, 3)
> g <- rbind(tp %x% temp1, tp %x% temp2)
```

Now we have the two dimensional LLTM design for three time points. The next thing is to stack as many of these designs as we have groups (here two) unto each other

```
> gr <- rep(1, 2)
> ipardes <- gr %x% g
```

With this the setup of the design for the item parameters is completed. We now must specify the treatment and trend effects.

We define a pseudodesign for treatment and time effects for T_2 and T_3 in the treatment group,

```
> eff1 <- matrix(c(rep(0, 4), c(1, 0, 1, 0), c(1, 1, 1,
+     1)), ncol = 4, nrow = 3, byrow = TRUE)
> eff1
```

```
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    1    0    1    0
[3,]    1    1    1    1
```

These effects have to be present for all 3 items of the 2 dimensions

```
> efftgd1 <- eff1 %x% c(1, 1, 1)
> efftg <- c(1, 1) %x% efftgd1
```

The same procedure has to be repeated for the control group, the only difference is that the pseudodesign lacks entries for treatment effects

```
> eff2 <- matrix(c(rep(0, 4), c(0, 0, 1, 0), c(0, 0, 1,
+     1)), ncol = 4, nrow = 3, byrow = TRUE)
> eff2
```

```
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    1    0
[3,]    0    0    1    1
```

```
> effcgd2 <- eff2 %x% c(1, 1, 1)
> effcg <- c(1, 1) %x% effcgd2
```

Now we must combine the design matrices for treatment and control group via stacking

```
> effdes <- rbind(efftg, effcg)
```

and finally combine the design matrix for item parameters with the design of the treatment and trend effects,

```
> des <- cbind(ipardes, effdes)
> effnam <- c("I2", "I3", "I5", "I6", "TreatEff2", "TreatEff3",
+     "TAU1", "TAU2")
> colnames(des) <- effnam
> des
```

```
      I2 I3 I5 I6 TreatEff2 TreatEff3 TAU1 TAU2
 [1,]  0  0  0  0         0         0    0    0
 [2,]  1  0  0  0         0         0    0    0
 [3,]  0  1  0  0         0         0    0    0
 [4,]  0  0  0  0         1         0    1    0
 [5,]  1  0  0  0         1         0    1    0
 [6,]  0  1  0  0         1         0    1    0
 [7,]  0  0  0  0         1         1    1    1
 [8,]  1  0  0  0         1         1    1    1
 [9,]  0  1  0  0         1         1    1    1
[10,]  0  0  0  0         0         0    0    0
[11,]  0  0  1  0         0         0    0    0
[12,]  0  0  0  1         0         0    0    0
```

```
[13,]  0  0  0  0         1         0    1    0
[14,]  0  0  1  0         1         0    1    0
[15,]  0  0  0  1         1         0    1    0
[16,]  0  0  0  0         1         1    1    1
[17,]  0  0  1  0         1         1    1    1
[18,]  0  0  0  1         1         1    1    1
[19,]  0  0  0  0         0         0    0    0
[20,]  1  0  0  0         0         0    0    0
[21,]  0  1  0  0         0         0    0    0
[22,]  0  0  0  0         0         0    1    0
[23,]  1  0  0  0         0         0    1    0
[24,]  0  1  0  0         0         0    1    0
[25,]  0  0  0  0         0         0    1    1
[26,]  1  0  0  0         0         0    1    1
[27,]  0  1  0  0         0         0    1    1
[28,]  0  0  0  0         0         0    0    0
[29,]  0  0  1  0         0         0    0    0
[30,]  0  0  0  1         0         0    0    0
[31,]  0  0  0  0         0         0    1    0
[32,]  0  0  1  0         0         0    1    0
[33,]  0  0  0  1         0         0    1    0
[34,]  0  0  0  0         0         0    1    1
[35,]  0  0  1  0         0         0    1    1
[36,]  0  0  0  1         0         0    1    1
```

Note that the first item of each dimension (item 1 and item 4 respectively) were fixed to be zero to allow estimation. Also we assume the treatment and trend effects to be the *same* for *both* dimensions. If we did not, we would have to specified a LLTM for each dimension seperately.

Now we are able to estimate that model with means of the **eRm** package. We only have to bear in mind that we reconstructed the data in such a way that we pretend to have groups × dimensions × time points different measurement points (here 12), so we specify

```
> mod <- LPCM(data, des, mpoints = 12)
> mod

Results of LPCM estimation:

Call:  LPCM(X = data, W = des, mpoints = 12)

Conditional log-likelihood: -380.4604
Number of iterations: 19
Number of parameters: 8

Basic Parameters eta:
                 I2          I3         I5         I6 TreatEff2
Estimate -1.0453137 -0.08779293 -2.4968011 -1.3851529 0.4446747
Std.Err   0.2386214  0.24198675  0.3134087  0.3069566 0.3639945
          TreatEff3      TAU1       TAU2
Estimate -0.3168535 0.8841091 -0.2296952
Std.Err   0.3679615 0.2769994  0.2770660
```

One could now compare (e.g. with the LR test) these treatment and trend effect estimates with the ones obtained from analyses with separate effects for each dimension to see if the generalisation is actually an apt simplification.

2. Polytomous Responses

The same ideas can also be used for the analysis of polytomous items where the number of categories and the category distances may be the same for all items (RSM) or different (PCM). The increased complexity of the models is reflected in a somewhat more sophisticated setup of the design matrix. However, all basic rules as discussed in the previous chapter still apply. We can even use the structure of the design matrices discussed so far. The main difference concerns the inclusion of the polytomous response categories.

2.1. The partial credit approach

For two time points, the model is

$$P(X_{vih1} = 1|T_1) = \frac{\exp(h\theta_{vi} + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta_{vi} + \omega_{il})}\,, \tag{4}$$

$$P(X_{vih2} = 1|T_2) = \frac{\exp(h\theta'_{vi} + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta'_{vi} + \omega_{il})} = \frac{\exp(h(\theta_{vi} + \delta_{vi}) + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta_{vi} + \delta_{vi} + \omega_{il})}\,, \tag{5}$$

where the parameters are defined as in (1), (2), and (3), h denotes the hth response category ($h = 0, \ldots, m_i$), $m_i + 1$ is the number of categories for item i, and ω_{ih} is the category parameter for item i.

Since in all models the first category is set to zero we always have to consider only m_i response categories (in case of dichotomous models there is only one category left being considered, $m_i = m = 1$, and therefore there was no need to include the categories in the design matrices so far).

2.1.1. All items with equal number of response categories

As an example, we specify a design matrix analogous to Table 4. We have a treatment and a control group and want to estimate a treatment effect λ_i and a trend effect τ_i for each item i. The items have 4 response categories ($h = 0, \ldots, 3$). The corresponding design matrix is given in Table 10.

The comparison of Tables 4 and 10 shows two main differences. First of all we have to include category parameters ω which are normalised such that $\omega_{i0} = \omega_{i1} = 0$ to ensure estimability [3] . Secondly, when specifying treatment and trend effects the categories must also be taken into account, since now the general change to be modelled is $h\delta_{vi}$, cf. (5) and (2). Since the first category is set equal to zero ($h = 0$) it is used as a baseline. The h values reflect how often the subjects need to show "effort" to achieve a score higher than 0. For example, to score 3 instead of 0, the subjects have to pass the scores of 0, 1, and 2. Thus, each entry for the λs and the τs in Table 4 is expanded for the categories, i.e. 1

[3] Each of the latent dimensions in the model is measured by the same real item i repeatedly presented to subjects (if we do not assume generalisation of certain effects over more than one item). Therefore the number of response categories is the same for all t virtual items representing the corresponding latent dimension i (in fact, t repeated presentations of the same real item i). It is therefore natural to assume the same category distances for the real item i over all t measurement points (rating scale assumption within one real item). Consequently, the category parameters are normalised such that $\omega_{i0} = \omega_{i1} = 0$. In priciple, we could specify different thresholds over the t measurement points. The number of parameters, however, would increase dramatically.

TABLE 10. Design matrix for different treatment and trend parameters for each item for a treatment group (TG) and a control group (CG).

				λ_1	λ_2	λ_3	τ_1	τ_2	τ_3	ω_{12}	ω_{13}	ω_{22}	ω_{23}	ω_{32}	ω_{33}
T_1	Item 1	TG	Cat 1												
			Cat 2							1					
			Cat 3								1				
		CG	Cat 1												
			Cat 2							1					
			Cat 3								1				
	Item 2	TG	Cat 1												
			Cat 2									1			
			Cat 3										1		
		CG	Cat 1												
			Cat 2									1			
			Cat 3										1		
	Item 3	TG	Cat 1												
			Cat 2											1	
			Cat 3												1
		CG	Cat 1												
			Cat 2											1	
			Cat 3												1
T_2	Item 1	TG	Cat 1	1			1								
			Cat 2	2			2			1					
			Cat 3	3			3				1				
		CG	Cat 1				1								
			Cat 2				2			1					
			Cat 3				3				1				
	Item 2	TG	Cat 1		1			1							
			Cat 2		2			2				1			
			Cat 3		3			3					1		
		CG	Cat 1					1							
			Cat 2					2				1			
			Cat 3					3					1		
	Item 3	TG	Cat 1			1			1						
			Cat 2			2			2					1	
			Cat 3			3			3						1
		CG	Cat 1						1						
			Cat 2						2					1	
			Cat 3						3						1

to m_i, in Table 10. The amount of "effort" needed to change from one category $(h-1)$ to the next (h) for item i is reflected by the values of ω_{ih}.

If there are more than two time points, the specifications follow those of Section 1.3. However, a main difference is that entries which would have been used in the design for a dichotomous model have to be multiplied by the category number (1 to m_i). For instance, the entry 2 in `dsgn3` on page 14 specifying the linear trend at T_3 for item 2 would result in `2 4 6` instead of `1 2 3` (cf. Example 4).

To complete the definition of design structures for fitting such models in **eRm**, again an item assignment vector has to be supplied. The specification is the same as described in Section 1.3.

Example 6: A treatment group ($n = 30$) and a control group ($n = 30$) have responded to 3 items (each with 4 response categories) at 2 time points. We want to estimate a trend effect and a treatment effect for each item.

The commands for reading the data (from file `llra_ex3.dat`) and modify them for an LLRA structure are

```
> dat3 <- read.table("llra_ex3.dat")
> data3 <- matrix(unlist(dat3), nc = 2)
```

The design matrix can either be specified using a null matrix (filled with 0s) with appropriate dimension and `fix()` to enter the corresponding numbers, or to piece it together from some submatrices. We will illustrate the latter method since it is easier for larger design matrices. We start defining a pseudodesign for treatment and time effects for T_2,

```
> pseudodes <- matrix(c(1, 0, 1, 1), 2, 2)
> rownames(pseudodes) <- c("TreatGroup", "CtrlGroup")
> colnames(pseudodes) <- c("Treatment", "Trend")
> pseudodes
```

```
           Treatment Trend
TreatGroup         1     1
CtrlGroup          0     1
```

Then we use a diagonal matrix representing the items, apply a Kronecker product to expand the pseudodesign,

```
> des0 <- diag(3) %x% pseudodes
```

and rearrange the columns for readability

```
> des0 <- des0[, c(1, 3, 5, 2, 4, 6)]
> effnam <- c("TreatEff1", "TreatEff2", "TreatEff3", "TAU1",
+     "TAU2", "TAU3")
> colnames(des0) <- effnam
> des0
```

```
     TreatEff1 TreatEff2 TreatEff3 TAU1 TAU2 TAU3
[1,]         1         0         0    1    0    0
[2,]         0         0         0    1    0    0
[3,]         0         1         0    0    1    0
[4,]         0         0         0    0    1    0
[5,]         0         0         1    0    0    1
[6,]         0         0         0    0    0    1
```

This is the treatment and trend effect structure for T_2. Since at T_1 all effects are at their baseline, the structure for T_1 is simply a null matrix of the same dimension as `des0` for T_2. Accordingly, we can use

```
> des0 <- c(0, 1) %x% des0
```

to obtain the design which would be used for a dichotomous model (cf. Example 2). Next we have to introduce the categories.

```
> des1 <- des0 %x% c(1, 2, 3)
```

This completes the setup for the treatment and trend effects. For polytomous models we additionally need covariates for the category parameters, which may be obtained in a similar way

```
> c0 <- matrix(c(0, 1, 0, 0, 0, 1), 3, 2)
> c1 <- c(1, 1) %x% c0
> c2 <- diag(3) %x% c1
```

Putting everything together gives the design matrix as shown in Table 10.

```
> des2 <- cbind(des1, rbind(c2, c2))
> colnames(des2) <- c(effnam, "C1.2", "C1.3", "C2.2", "C2.3",
+     "C3.2", "C3.3")
```

Finally, we need the assignment vector (2 groups $\times$ 3 items)

```
> grpspoly <- as.numeric(gl(6, 30))
```

The model is

```
> res.lpcm <- LPCM(data3, des2, mpoints = 2, groupvec = grpspoly)
> res.lpcm

Results of LPCM estimation:

Call:  LPCM(X = data3, W = des2, mpoints = 2, groupvec = grpspoly)

Conditional log-likelihood: -132.6720
Number of iterations: 43
Number of parameters: 12

Basic Parameters eta:
         TreatEff1  TreatEff2 TreatEff3      TAU1      TAU2     TAU3
Estimate 1.4689254 -0.7297481 0.4609431 0.2501465 1.5374481 1.311026
Std.Err  0.5812815  0.4797781 0.5008556 0.3572245 0.4241286 0.420396
               C1.2      C1.3      C2.2       C2.3       C3.2
Estimate -1.2678423 -4.163966 -1.194317 -3.3847270 -1.6196304
Std.Err   0.4841956  1.111025  0.469731  0.9935174  0.5243751
              C3.3
Estimate -4.377936
Std.Err   1.067895
```

Inspection of the parameter estimates shows a significant positive trend for items 2 and 3, the respondents tend to choose higher categories at T_2 compared to T_1. A treatment effect can only be observed for item 1.

Once again, hypotheses about generalisability of trend or treatment effects can be investigated by collapsing the appropriate columns, fitting these models and using the likelihood ratio test.

2.1.2. **Items with different number of response categories**

With the partial credit approach it is also possible to estimate trend, treatment and category effects if the number of categories differ across items. In Example 6 all items had the same number of categories, but it is possible to simultaneously analyse items with different numbers of categories, for example, a questionnaire with dichotomous and polytomous items.

Example 7: A treatment group ($n = 30$) and a control group ($n = 30$) have responded to 3 items at 2 time points. Item 1 has 3 categories, item 2 is dichotomous and item 3 has 4 categories. We want to estimate a trend effect and a treatment effect for each item.

The commands for reading the data (from file `llra_ex3_a.dat`) and rearrange them for an LLRA structure are

```
> dat3a <- read.table("llra_ex3_a.dat")
> data3a <- matrix(unlist(dat3a), nc = 2)
```

These data are a modified version of the data used in the previous example. The data for item 3 remain the same but the data for the other two items have been altered such that the higher categories have been merged into one. Specifically, for item 1, category 2 contains all category 2 and 3 responses from the original data and for item 2, category 1 contains all non-zero categories. Again there are several ways to set up the design matrix. Either by defining a null matrix of appropriate dimensions and by using `fix()` to enter the corresponding numbers. Or to build it up from some submatrices.

The easiest way to do it is to use the design matrix from 6 and modify it accordingly. This is again the (summarised) code to build up the design matrix with the same number of categories for all items (as previously discussed in Example 6), now stored into `des3`.

```
> pseudodes <- matrix(c(1, 0, 1, 1), 2, 2)
> rownames(pseudodes) <- c("TreatGroup", "CtrlGroup")
> colnames(pseudodes) <- c("Treatment", "Trend")
> des0 <- diag(3) %x% pseudodes
> des0 <- des0[, c(1, 3, 5, 2, 4, 6)]
> effnam <- c("TreatEff1", "TreatEff2", "TreatEff3", "TAU1",
+     "TAU2", "TAU3")
> colnames(des0) <- effnam
> des0 <- c(0, 1) %x% des0
> des1 <- des0 %x% c(1, 2, 3)
> c0 <- matrix(c(0, 1, 0, 0, 0, 1), 3, 2)
> c1 <- c(1, 1) %x% c0
> c2 <- diag(3) %x% c1
> des3 <- cbind(des1, rbind(c2, c2))
> colnames(des3) <- c(effnam, "C1.2", "C1.3", "C2.2", "C2.3",
+     "C3.2", "C3.3")
```

Consider a design matrix $\mathbf{W}$ for items with the same number of categories, and a design matrix $\mathbf{W}^*$ for items with different number of categories. If $\mathbf{W}$ is such that the number of categories is the same as for the item(s) with the maximal number of categories in $\mathbf{W}^*$, then the dimension of $\mathbf{W}$ must be larger than $\mathbf{W}^*$. To obtain $\mathbf{W}^*$ from $\mathbf{W}$ the columns that represent the nonexisting categories in $\mathbf{W}^*$ must be deleted from $\mathbf{W}$. For our examples, $\mathbf{W}$ relates to the design matrix `des2` from Example 6 and $\mathbf{W}^*$ to the design

matrix for Example 7. We delete the columns corresponding to the superfluous category parameters to ensure full column rank

```
> des3a <- des3[, -(8:10)]
```

Actually, the corresponding rows should be deleted as well. However, when using `eRm` for LLRAs it is not possible (for technical reasons) to delete these rows from the design matrix **W** (as would be done if a simple PCM was used for scale analysis). The number of rows of the design matrix must be the same in both cases. Instead of deletion, the superfluous category rows are filled with 0s.

```
> des3a[c(3, 6, 8, 9, 11, 12, 21, 24, 26, 27, 29, 30),
+     ] <- 0
```

This is the complete setup for the design matrix. It is important to keep in mind that a design matrix with equal number of categories for every item has to be set up first and that this number must be the maximum number of item categories.

Finally, we once more need the item assignment vector (which is the same for Example 6 and Example 7)

```
> grpspoly <- as.numeric(gl(6, 30))
```

The model is fitted by

```
> res.lpcm2 <- LPCM(data3a, des3a, mpoints = 2, groupvec = grpspoly)
> res.lpcm2

Results of LPCM estimation:

Call:  LPCM(X = data3a, W = des3a, mpoints = 2, groupvec = grpspoly)

Conditional log-likelihood: -101.261
Number of iterations: 32
Number of parameters: 9

Basic Parameters eta:
         TreatEff1  TreatEff2 TreatEff3      TAU1      TAU2      TAU3
Estimate 1.3750292 -0.4448694 0.4609519 0.5751396 3.1237565 1.3110184
Std.Err  0.5481639  0.9551232 0.5008556 0.3110730 0.7340798 0.4203942
                C1.2       C3.2      C3.3
Estimate -0.03226568 -1.6196293 -4.377925
Std.Err   0.40979976  0.5243734  1.067891
```

Compared to the results of Example 6, the parameter estimates still show significant positive trends for items 2 and 3, the respondents tend to choose higher categories at T_2 than at T_1. A treatment effect can again only be observed for item 1. It should be noted that the estimates for item 3 do not really change, because it was the item that had the same number of categories as in Example 6.

Again, results for specific hypotheses on generalisability of treatment or trend effects over several items are obtained by collapsing appropriate columns and fitting that model. Another possible simplification concerns the scaling of the categories. This is the topic of the next section.

2.2. **The rating scale approach**

If we assume the category distances to be the same across all items (*equidistant scoring*), then model (5) for T_2 simplifies to

$$P(X_{vih2} = 1|T_2) = \frac{\exp(h(\theta_{vi} + \delta_{vi}) + \omega_h)}{\sum_{l=0}^{m_i} \exp(l\theta_{vi} + \delta_{vi} + \omega_l)}\,, \tag{6}$$

where now $\omega_{ih} = \omega_h$ for all items. All other considerations of the previous section also apply to (6).

Example 8 (Example 6 continued)**:** We try to simplify the model using equal category parameters across the items.

Basically, to specify this simpler model we have to collapse columns 7, 9, 11, and 8, 10, 12, respectively.

```
> cat2 <- rowSums(des2[, c(7, 9, 11)])
> cat3 <- rowSums(des2[, c(8, 10, 12)])
```

and add the new columns to the design for treatment and trend effects `des1`.

```
> des3 <- cbind(des1, cat2, cat3)
> colnames(des3) <- c(effnam, "C.2", "C.3")
```

Fitting this simpler model gives

```
> res.lrsm <- LPCM(data3, des3, mpoints = 2, groupvec = grpspoly)
> res.lrsm

Results of LPCM estimation:

Call:  LPCM(X = data3, W = des3, mpoints = 2, groupvec = grpspoly)

Conditional log-likelihood: -133.1761
Number of iterations: 20
Number of parameters: 8

Basic Parameters eta:
         TreatEff1  TreatEff2 TreatEff3      TAU1      TAU2      TAU3
Estimate  1.339879 -0.7904490 0.4295578 0.2461319 1.6843826 1.2327487
Std.Err   0.513233  0.4914142 0.4831179 0.3523761 0.3955073 0.3590487
               C.2       C.3
Estimate -1.3510798 -3.917679
Std.Err   0.2830229  0.601310
```

A likelihood ratio test to evaluate if the simplification is admissable shows that we do not need to model item specific category parameters.

```
> lrtst(res.lpcm, res.lrsm)

Likelihood ratio statistic: 1.008248 df = 4 p = 0.909
```

Again, other hypotheses (like generalisation of treatment or trend effects across several items) can be specified by deleting appropriate rows and columns of the design matrix in analogy to previous examples.

2.3. More than two time points

The general formulation of the relaxed assumption model for arbitrary time points is

$$(7) \qquad P(X_{viht} = 1|T_t) = \frac{\exp(h\theta'_{vi} + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta'_{vi} + \omega_{il})} = \frac{\exp(h(\theta_{vi} + \delta_{vit}) + \omega_{ih})}{\sum_{l=0}^{m_i} \exp(l\theta_{vi} + \delta_{vit} + \omega_{il})} ,$$

where

$$(8) \qquad \delta_{vit} = \mathbf{w}_{it}^T \boldsymbol{\eta} ,$$

and $\mathbf{w}_{it}^T$ is a row in the design matrix specifying certain effects on (real) item i at time t for subject (or all subjects of treatment group) v.

Example 9: A treatment group ($n = 30$) and a control group ($n = 30$) have been observed at 3 time points. The study design is

	T_1	T_2	T_3
TG	Baseline	Treatment	–
CG	Baseline	–	–

The items with 4 categories measure the severity of three symptoms (the first category represents highest severity) at each time point. The question is whether the treatment is effective at T_2 and if this effect is still observable after a period without treatment at T_3. We assume the same category differences for all items (rating scale approach). We want to estimate a model with a treatment parameters λ for every item comparing T_1 and T_2 and T_1 and T_3 (i.e., 6 treatment parameters) one general trend parameter τ.

The data are in file `llra_ex4.dat`. We first read the data and modify them for the LLRA structure. The first 30 cases belong to the treatment group, the other 30 are the controls.

```
> data4 <- read.table("llra_ex4.dat", header = FALSE)
> dat4 <- matrix(unlist(data4), nc = 3)
```

The pseudodesign is given in Table 11.

We start to construct the design matrix using `d1m` which corresponds to the final setup for the τs at T_2. The other specifications are analogous to Example 4.

```
> d1 <- c(1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
+     0, 1, 0)
> d1m <- matrix(d1 %x% c(1, 2, 3), 18)
> d2m <- diag(2) %x% d1m
> trnd <- rep(1:3, 2 * 3 * 2)
> trnd <- trnd * as.numeric(gl(2, 3 * 2 * 3))
> d3m <- cbind(d2m, trnd)
> design <- rbind(matrix(0, 18, 7), d3m)
```

TABLE 11. Pseudo design matrix (neglecting items and categories) for effect parameters between T_1 and T_2 ($\lambda^{t_2-t_1}$), and between T_1 and T_3 ($\lambda^{t_3-t_1}$) and a general trend parameter τ.

		$\lambda^{t_2-t_1}$	$\lambda^{t_3-t_1}$	τ
T_1	TG			
	CG			
T_2	TG	1		1
	CG			1
T_3	TG		1	1
	CG			1

```
> cat2 <- rep(c(0, 1, 0), 3 * 3 * 2)
> cat3 <- rep(c(0, 0, 1), 3 * 3 * 2)
> design <- cbind(design, cat2, cat3)
> colnames(design) <- c("L.1(2)", "L.2(2)", "L.3(2)", "L.1(3)",
+     "L.2(3)", "L.3(3)", "TREND", "C.2", "C.3")
> groups <- as.numeric(gl(6, 30))
```

Fitting the model gives

```
> res.ex4 <- LPCM(dat4, design, mpoints = 3, groupvec = groups)
> res.ex4

Results of LPCM estimation:

Call:  LPCM(X = dat4, W = design, mpoints = 3, groupvec = groups)

Conditional log-likelihood: -300.3035
Number of iterations: 50
Number of parameters: 9

Basic Parameters eta:
           L.1(2)    L.2(2)    L.3(2)     L.1(3)    L.2(3)    L.3(3)
Estimate 1.0301068 0.4611837 1.2037938 -0.2398956 1.2096525 2.6403020
Std.Err  0.3464313 0.3323187 0.3680473  0.3655568 0.4091801 0.5267083
             TREND        C.2        C.3
Estimate 0.40981701 -1.3353425 -3.7098060
Std.Err  0.09808928  0.2143427  0.4252225
```

In general, there is a positive trend for both groups over the whole observation period. When comparing treatment and control group, immediately after the therapy at T_2 there is significant difference in improvement for symptoms 1 (`L.1(2)`) and 3 (`L.3(2)`), but not for symptom 2. Comparing the improvement between T_1 and T_3, the difference between treatment and control group is still observable for symptom 3 (`L.3(3)`) and now also for symptom 2 (`L.2(3)`) whereas the difference with regard to symptom 1 (`L.1(3)`) has vanished.

2.4. **Modelling change over unidimensional subscales**

Again, the methods described can be used to estimate generalised effects over different latent dimensions each of which is measured with unidimensional items. This is basically

analogous to Section 1.4, only in case of polytomous items we have to consider the number of categories in our design matrix, which means we use multidimensional LRSMs oder LPCMs for our purpose. The only difference lies within the number of categories and their respective entries in the design matrix.

Let us again assume w.l.o.g that at each time point, the items are ordered according to the dimension they measure, e.g. the first $s1$ items measure dimension 1, the next $s2$ items measure dimension 2 and so on. The responses are polytomous, with $m_i + 1$ categories $h = 0, \ldots, m_i$. If we denote with X_{gdt} the matrix of responses of all people S_v within group g to the items measuring dimension d at time t, we have the usual data structure of repeated measurements if only looking on dimensions and groups as in Table 7.

To apply the multidimensional model described above, that data structure must be rearranged in exactly the same fashion as in Table 8.

The only practical difference to Section 1.4 and between LRSM and LPCM lies in the specification of the design matrix. It has to be changed to include entries for the categories as well, the same way as it has been done in the previous Subsections. For a LRSM we assume all categories to be the same for each item within each dimension but in the LPCM framework we allow the categories to differ, so every item has its own categories. The setup of the design matrix is as follows

$$\text{(Treatment) Groups} \sqsubset \text{Dimensions} \sqsubset \text{Time Points} \sqsubset \text{Categories}$$

Table 12 shows how the design matrix for a single treatment group and one dimension has to be constructed. For every dimension the matrix has to be expanded accordingly and for every group such a matrix has to be specified and they must all be stacked upon each other. The big difference to Table 9 is that we blow up the design matrix to incorporate the categories and add as many columns as there are category parameters minus one for each dimension.

TABLE 12. Design matrix for multidimensional LPCM for a single treatment group and one dimension.

				β_2	$\dots$	β_{s1}	$\beta_{(\sum_d (sd-1)+1)}$	$\dots$	$\dots$	$\beta_{(\sum_d sd)}$	$\dots$	λ_d	$\dots$	τ_d	$\dots$	$c_2 D_d$	$\dots$	$c_{m_i} D_d$	$\dots$
G_1	D_d	T1	C_1				0									0	$\dots$	0	
			$\vdots$				$\vdots$									1	$\dots$	0	
			C_{m_i}				0									0	$\dots$	1	
			C_1					1								0	$\dots$	0	
			$\vdots$					$\vdots$								1	$\dots$	0	
			C_{m_i}					m_i								0	$\dots$	1	
			$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
			$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
			$C1$							1						0	$\dots$	0	
			$\vdots$							$\vdots$						1	$\dots$	0	
			C_{m_i}							m_i						0	$\dots$	1	
		T2	$C1$				0					1		1		0	$\dots$	0	
			$\vdots$				$\vdots$					$\vdots$		$\vdots$		1	$\dots$	0	
			C_{m_i}				0					m_i		m_i		0	$\dots$	1	
			$C1$					1				1		1		0	$\dots$	0	
			$\vdots$					$\vdots$				$\vdots$		$\vdots$		1	$\dots$	0	
			C_{m_i}					m_i				m_i		m_i		0	$\dots$	1	
			$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
			$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
			C_1							1		1		1		0	$\dots$	0	
			$\vdots$							$\vdots$		$\vdots$		$\vdots$		1	$\dots$	0	
			C_{m_i}							m_i		m_i		m_i		0	$\dots$	1	

This setup can be used for LPCMs and LRSMs alike, only that in LRSM we consider m category parameters for each dimension, whereas with the LPCM we have to specify different categories for every item.

Example 10: A treatment group ($n = 50$) and a control group ($n = 30$) have responded to 6 items at 3 time points. W.l.o.g, the first 3 items measure one and the same latent dimension, items 4 to 6 measure a second dimension. We want to estimate a trend effect and a treatment effect that generalises over all dimensions, assuming unidimensional measurement within each dimension. This time, the items have four categories and we assume the categories to be constant within each dimension (i.e. a LRSM approach).

The data are given in the file `hyblrsm.dat`. To rearrange the data matrix, we must use the same code as in Section 1.4.

```
> dats <- as.matrix(read.table("hyblrsm.dat", header = TRUE))
```

To get the design matrix for the LRSM we best use the matrix from Example 5, which we called `des`.

```
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    1    0
[3,]    0    0    1    1
```

We can blow up that design matrix with the number of categories by using

```
> desRSM <- des %x% c(1, 2, 3)
```

which gives us the design for the item parameters and the effect parameters. The matrix already has groups × time points × dimensions × categories rows. To add the additionally needed category parameters, we can use

```
> cat2 <- c(0, 1, 0)
> cat3 <- c(0, 0, 1)
```

to construct the category contrasts and

```
> d1 <- c(rep(1, 9), rep(0, 9), rep(1, 9), rep(0, 9))
> d2 <- c(rep(0, 9), rep(1, 9), rep(0, 9), rep(1, 9))
```

to get indicators for each dimension. We need one column for dimension 1 and category 2, one column for dimension 1 and category 3 and so on.

```
> c2d1 <- d1 %x% cat2
> c2d2 <- d2 %x% cat2
> c3d1 <- d1 %x% cat3
> c3d2 <- d2 %x% cat3
```

Combining these columns and adding them to `desRSM` yields the desired design matrix for the generalised effects over subscales

```
> descat <- cbind(c2d1, c3d1, c2d2, c3d2)
> des <- cbind(desRSM, descat)
> effnam <- c("I2", "I3", "I5", "I6", "TreatEff2", "TreatEff3",
+     "TAU1", "TAU2", "c2d1", "c3d1", "c2d2", "c3d2")
```

```
> colnames(des) <- effnam
> head(des)

     I2 I3 I5 I6 TreatEff2 TreatEff3 TAU1 TAU2 c2d1 c3d1 c2d2 c3d2
[1,]  0  0  0  0         0         0    0    0    0    0    0    0
[2,]  0  0  0  0         0         0    0    0    1    0    0    0
[3,]  0  0  0  0         0         0    0    0    0    1    0    0
[4,]  1  0  0  0         0         0    0    0    0    0    0    0
[5,]  2  0  0  0         0         0    0    0    1    0    0    0
[6,]  3  0  0  0         0         0    0    0    0    1    0    0
```

Now we are able to estimate the model with means of the **eRm** package. We only have to bear in mind that we reconstructed the data in such a way that we pretend to have groups × dimensions × time points different measurement points (here 12), so we specify

```
> mod <- LPCM(data, des, mpoints = 12)
> mod

Results of LPCM estimation:

Call:  LPCM(X = data, W = des, mpoints = 12)

Conditional log-likelihood: -1034.502
Number of iterations: 37
Number of parameters: 12

Basic Parameters eta:
                 I2        I3         I5         I6 TreatEff2  TreatEff3
Estimate -0.3902630 0.1183652 -0.1741263 -0.7935223 0.6653960 -0.3123125
Std.Err   0.1079259 0.1090176  0.1100458  0.1201562 0.1631708  0.1585161
             TAU1       TAU2      c2d1      c3d1       c2d2       c3d2
Estimate 0.3548800 0.05918625 0.4178096 0.6614449 -0.2214178 -0.5807974
Std.Err  0.1232122 0.12168708 0.2493343 0.3990607  0.2161424  0.3553549
```

One could now compare (e.g. with the LR test) these treatment and trend effect estimates with the ones obtained from analyses with separate effects for each dimension.

To estimate a LPCM, additional columns for different category parameters for each item have to be added, as explained in the former sections.

References

Fischer, G. H. (1989). An IRT-based model for dichotomous longitudinal data. *Psychometrika*, 54:599–624.

Fischer, G. H. (1995). Linear logistic models for change. In Fischer, G. and Molenaar, I., editors, *Rasch Models: Foundations, Recent Developments, and Applications*, pages 157–180. Springer, New York.

Mair, P. and Hatzinger, R. (2008). *eRm: Extended Rasch Modeling.* R package version 0.10-1.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Appendix A. Basics of Kronecker products

If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ matrix, then the Kronecker product $\mathbf{A} \otimes \mathbf{B}$ is the $mp \times nq$ block matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}.$$

In simple terms, the result has the same structure as the left hand side, but *each element of the left* hand side is *blown up by the whole right* hand side.

Four simple examples:

Let $\mathbf{a}$ be a column vector with 2 elements and $\mathbf{B}$ a 2×2 matrix

$$\mathbf{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 3 \\ 5 & 7 \end{bmatrix}.$$

The four examples are:

$$\mathbf{a} \otimes \mathbf{B} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 0 & 3 \\ 5 & 7 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ 5 & 7 \\ 0 & 6 \\ 10 & 14 \end{bmatrix}$$

$$\mathbf{a}^T \otimes \mathbf{B} = \begin{bmatrix} 1 & 2 \end{bmatrix} \otimes \begin{bmatrix} 0 & 3 \\ 5 & 7 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 0 & 6 \\ 5 & 7 & 10 & 14 \end{bmatrix}$$

$$\mathbf{B} \otimes \mathbf{a} = \begin{bmatrix} 0 & 3 \\ 5 & 7 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ 0 & 6 \\ 5 & 7 \\ 10 & 14 \end{bmatrix}$$

$$\mathbf{B} \otimes \mathbf{a}^T = \begin{bmatrix} 0 & 3 \\ 5 & 7 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 3 & 6 \\ 5 & 10 & 7 & 14 \end{bmatrix}$$

In **R** these can be obtained by

```
> a <- c(1, 2)
> B <- matrix(c(0, 3, 5, 7), ncol = 2, nrow = 2, byrow = TRUE)
> a %x% B
> t(a) %x% B
> B %x% a
> B %x% t(a)
```

Remarks: The definition of `a` by using the *combine* function `c()` results in `a` a column vector `a`. This is not so obvious when we print `a` which results in `[1] 1 2`. But using the transpose function `t()` twice gives the "correct" display, i.e., `t(t(a))`. For the definition of `B` we used the `byrow = TRUE` option to illustrate that he matrix should be filled by rows and not by columns (the default).