Package 'immer'

March 21, 2024

Type Package

Title Item Response Models for Multiple Ratings

Version 1.5-13

Date 2024-03-21 15:57:11

Author Alexander Robitzsch [aut, cre], Jan Steinfeld [aut]

Maintainer Alexander Robitzsch <robitzsch@ipn.uni-kiel.de>

Description Implements some item response models for multiple ratings, including the hierarchical rater model, conditional maximum likelihood estimation of linear logistic partial credit model and a wrapper function to the commercial FACETS program. See Robitzsch and Steinfeld (2018) for a description of the functionality of the package. See Wang, Su and Qiu (2014; <doi:10.1111/jedm.12045>) for an overview of modeling alternatives.

Depends R (>= 3.0-0)

Imports CDM, coda, graphics, methods, psychotools, Rcpp, sirt, stats, TAM

Suggests MASS

Enhances eRm, lme4

LinkingTo Rcpp, RcppArmadillo

URL https://github.com/alexanderrobitzsch/immer,
https://sites.google.com/site/alexanderrobitzsch2/software

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-03-21 15:20:02 UTC

R topics documented:

`immer-package` *Item Response Models for Multiple Ratings*

Description

Implements some item response models for multiple ratings, including the hierarchical rater model, conditional maximum likelihood estimation of linear logistic partial credit model and a wrapper function to the commercial FACETS program. See Robitzsch and Steinfeld (2018) for a description of the functionality of the package. See Wang, Su and Qiu (2014; <doi:10.1111/jedm.12045>) for an overview of modeling alternatives.

Details

The **immer** package has following features:

- Estimation of the hierarchical rater model (Patz et al., 2002) with `immer_hrm` and simulation of it with `immer_hrm_simulate`.
- The linear logistic partial credit model as an extension to the linear logistic test model (LLTM) for dichotomous data can be estimated with conditional maximum likelihood (Andersen, 1995) using `immer_cml`.
- The linear logistic partial credit model can be estimated with composite conditional maximum likelihood (Varin, Reid & Firth, 2011) using the `immer_ccml` function.
- The linear logistic partial credit model can be estimated with a bias-corrected joint maximum likelihood method (Bertoli-Barsotti, Lando & Punzo, 2014) using the `immer_jml` function.

- Wrapper function `immer_FACETS` to the commercial program FACETS (Linacre, 1999) for analyzing multi-faceted Rasch models.
- ...

Author(s)

Alexander Robitzsch [aut, cre], Jan Steinfeld [aut]

Maintainer: Alexander Robitzsch <robitzsch@ipn.uni-kiel.de>

References

Andersen, E. B. (1995). Polytomous Rasch models and their estimation. In G. H. Fischer & I. W. Molenaar (Eds.). *Rasch Models* (pp. 39-52). New York: Springer.

Bertoli-Barsotti, L., Lando, T., & Punzo, A. (2014). Estimating a Rasch Model via fuzzy empirical probability functions. In D. Vicari, A. Okada, G. Ragozini & C. Weihs (Eds.). *Analysis and Modeling of Complex Data in Behavioral and Social Sciences*, Springer.

Linacre, J. M. (1999). *FACETS* (Version 3.17)[Computer software]. Chicago: MESA.

Patz, R. J., Junker, B. W., Johnson, M. S., & Mariano, L. T. (2002). The hierarchical rater model for rated test items and its application to large-scale educational assessment data. *Journal of Educational and Behavioral Statistics, 27*(4), 341-384.

Robitzsch, A., & Steinfeld, J. (2018). Item response models for human ratings: Overview, estimation methods, and implementation in R. *Psychological Test and Assessment Modeling, 60*(1), 101-139.

Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica, 21*, 5-42.

Wang, W. C., Su, C. M., & Qiu, X. L. (2014). Item response models for local dependence among multiple ratings. *Journal of Educational Measurement, 51*(3), 260-280.

See Also

For estimating the Rasch multi-facets model with marginal maximum likelihood see also the `TAM::tam.mml.mfr` and `sirt::rm.facets` functions.

For estimating the hierarchical rater model based on signal detection theory see `sirt::rm.sdt`.

For conditional maximum likelihood estimation of linear logistic partial credit models see the **eRm** (e.g. `eRm::LPCM`) and the **psychotools** (e.g. `psychotools::pcmodel`) packages.

Examples

```
##
##
##  immer 0.0-16 (2015-06-27)
##
##
##      #@####################################################################@
##     :#@####################################################################`
##     #######################################################################+
```

```
##     #####################################################################@
##    .##########@###########################################################`
##    +######@+;'@##########################################################@:
##    ######'     +@##########################################################
##    #####+ ##'  `#########################################################@
##    ####@ ####  +@#@##################################################@#####
##   .#####.#####@######@#########@#@######@@####@##@############@######@#####`
##   :####+:###@:,@##@,;##@+@##@+'###@;.'##@##@##@''@#####@@'+@###,;##@;######.
##   ;####@ @##   ##@  ,;    :,    .#    ':    :.    ,@@#'     ';   ,+   @####.
##   '#####`:##   #@   : ''    ++   @@;  ` +;    +'   #@' `+@   #@  .    #####,
##   +######, #   : ,   @#@   +#@  `##,   @#@   @##  `#@  `##   @#`  +,  @####:
##   +#######@@  :+@`  ;##'  `##@  .@#   +##:  `###  .@.  @#+   #:  .##@@#####:
##   '########:  ##@   @#@,  ;##;  ;#@   @##`  +##,  +#       `@#`  ##########,
##   ;########   @##   ###`  @##`  @#+  `###   @##   ##   ,;'@##@   @#########,
##   :#######@  `##+  ,##@   @#@   @#,  '##@   ##@   @@  `#######  `##########.
##   .#######+  ,##.  @##+  `##@  .##`  @##;  ,##@  `##   ###; #;  :#@,+######
##    #######'    #   @##:  ,##;  ;##   @##`  +###   `@:     `;#.  @@:  ######
##    #######@,`,##.`,###,..###,   #@#..,##@,..####:` +#@'  ``#+#` `@#+ :#####@
##    #########################:   ,###################@#####@###@#@#########@
##    '#######################@+;+###########################################:
##    ,######################################################################`
##     ##@##################################################################@
```

---

data.immer *Some Example Datasets for the* **immer** *Package*

Description

Some example rating datasets for the **immer** package.

Usage

```
data(data.immer01a)
data(data.immer01b)
data(data.immer02)
data(data.immer03)
data(data.immer04a)
data(data.immer04b)
data(data.immer05)
data(data.immer06)
data(data.immer07)
data(data.immer08)
data(data.immer09)
data(data.immer10)
data(data.immer11)
data(data.immer12)
```

Format

- The format of the dataset `data.immer01a` is:
  `'data.frame': 23904 obs. of 8 variables:`
  `$ idstud: int 10001 10001 10003 10003 10003 10004 10004 10005 10005 10006 ...`
  `$ type : Factor w/ 2 levels "E","I": 1 2 1 1 2 1 2 1 2 1 ...`
  `$ rater : Factor w/ 57 levels "R101","R102",..: 1 36 33 20 21 57 36 9 31 21 ...`
  `$ k1 : int 2 1 0 0 0 2 2 1 2 0 ...`
  `$ k2 : int 1 1 0 0 0 1 1 1 2 0 ...`
  `$ k3 : int 1 1 0 0 0 1 1 1 2 1 ...`
  `$ k4 : int 2 2 1 0 0 1 1 1 2 1 ...`
  `$ k5 : int 1 2 0 0 0 2 1 2 3 2 ...`

- The format of the dataset `data.immer01b` is:
  `'data.frame': 4244 obs. of 8 variables:`
  `$ idstud: int 10001 10003 10005 10007 10009 10016 10018 10022 10024 10029 ...`
  `$ type : Factor w/ 1 level "E": 1 1 1 1 1 1 1 1 1 1 ...`
  `$ rater : Factor w/ 20 levels "R101","R102",..: 1 20 9 5 14 19 20 6 10 10 ...`
  `$ k1 : int 2 0 1 2 2 2 3 1 3 2 ...`
  `$ k2 : int 1 0 1 2 2 1 3 2 2 1 ...`
  `$ k3 : int 1 0 1 1 3 2 2 1 3 1 ...`
  `$ k4 : int 2 0 1 2 3 2 2 2 3 2 ...`
  `$ k5 : int 1 0 2 1 3 1 2 3 3 1 ...`
  This dataset is a subset of `data.immer01a`.

- The format of the dataset `data.immer02` is:
  `'data.frame': 6105 obs. of 6 variables:`
  `$ idstud: int 10002 10004 10005 10006 10007 10008 10009 10010 10013 10014 ...`
  `$ rater : Factor w/ 44 levels "DR101","DR102",..: 43 15 12 21 9 3 35 24 11 17 ...`
  `$ a1 : int 3 1 2 1 0 2 1 2 1 1 ...`
  `$ a2 : int 3 0 3 1 0 3 0 2 2 1 ...`
  `$ a3 : int 1 2 0 1 2 3 2 2 1 1 ...`
  `$ a4 : int 2 1 2 1 1 3 1 2 2 1 ...`

- The format of the dataset `data.immer03` is:
  `'data.frame': 6466 obs. of 6 variables:`
  `$ idstud: int 10001 10002 10003 10004 10005 10006 10007 10009 10010 10012 ...`
  `$ rater : Factor w/ 44 levels "R101","R102",..: 18 10 8 25 19 31 16 22 29 6 ...`
  `$ b1 : int 1 2 1 3 3 2 3 2 2 1 ...`
  `$ b2 : int 2 1 0 3 3 1 1 2 2 1 ...`
  `$ b3 : int 2 3 1 2 3 1 2 2 2 2 ...`
  `$ b4 : int 1 2 0 2 2 2 3 2 3 1 ...`

- The format of the dataset `data.immer04a` is:
  `'data.frame': 25578 obs. of 7 variables:`
  `$ idstud: int 10001 10001 10001 10002 10002 10002 10003 10003 10004 10004 ...`
  `$ task : Factor w/ 4 levels "l1","l2","s1",..: 1 4 4 1 1 3 1 3 2 2 ...`

```
$ rater : Factor w/ 43 levels "R101","R102",..: 14 31 25 39 35 19 43 27 12 4 ...
$ TA : int 5 2 4 0 0 0 2 6 5 3 ...
$ CC : int 4 1 3 1 0 0 2 6 4 3 ...
$ GR : int 4 1 2 1 0 0 1 7 5 2 ...
$ VOC : int 4 2 3 1 0 0 1 6 5 3 ...
```

- The format of the dataset `data.immer04b` is:

```
'data.frame': 2975 obs. of 7 variables:
$ idstud: int 10002 10004 10010 10013 10015 10016 10024 10025 10027 10033 ...
$ task : Factor w/ 1 level "s1": 1 1 1 1 1 1 1 1 1 1 ...
$ rater : Factor w/ 20 levels "R101","R102",..: 19 1 5 16 13 13 8 10 19 5 ...
$ TA : int 0 3 5 5 3 2 3 6 4 5 ...
$ CC : int 0 3 4 5 4 1 4 7 3 3 ...
$ GR : int 0 3 3 6 5 2 3 6 3 2 ...
$ VOC : int 0 2 4 6 5 2 3 6 3 2 ...
```

  This dataset is a subset of `data.immer04a`.

- The format of the dataset `data.immer05` is:

```
'data.frame': 21398 obs. of 9 variables:
$ idstud : int 10001 10001 10002 10002 10003 10003 10004 10004 10005 10005 ...
$ type : Factor w/ 2 levels "l","s": 2 1 2 1 2 1 2 1 2 1 ...
$ task : Factor w/ 6 levels "l1","l4","l5",..: 5 2 6 3 5 1 5 1 5 2 ...
$ rater : Factor w/ 41 levels "ER101","ER102",..: 1 40 38 23 37 33 2 33 21 27 ...
$ idstud_task: Factor w/ 19484 levels "10001l4","10001s3",..: 2 1 4 3 6 5 8 7 10 9 ...
$ TA : int 3 4 6 6 4 2 0 3 1 3 ...
$ CC : int 5 4 5 5 3 3 0 2 5 3 ...
$ GR : int 4 4 5 6 5 3 0 4 5 4 ...
$ VO : int 6 4 6 6 4 3 0 3 4 3 ...
```

- The dataset `data.immer06` is a string containing an input syntax for the FACETS program.
- The format of the dataset `data.immer07` is:

```
'data.frame': 1500 obs. of 6 variables:
$ pid : int 1 1 1 2 2 2 3 3 3 4 ...
$ rater: chr "R1" "R2" "R3" "R1" ...
$ I1 : num 1 1 2 1 1 1 0 1 1 2 ...
$ I2 : num 0 1 1 2 1 2 1 1 2 1 ...
$ I3 : num 1 1 2 0 0 1 1 0 2 1 ...
$ I4 : num 0 0 1 0 0 1 0 1 2 0 ...
```

- The format of the dataset `data.immer08` (example in Schuster & Smith, 2006) is

```
'data.frame': 16 obs. of 3 variables:
$ Facility: int 1 1 1 1 2 2 2 2 3 3 ...
$ Research: int 1 2 3 4 1 2 3 4 1 2 ...
$ weights : int 40 6 4 15 4 25 1 5 4 2 ...
```

- The dataset `data.immer09` contains reviewer ratings for conference papers (Kuhlisch et al., 2016):

```
'data.frame': 128 obs. of 3 variables:
$ idpaper : int 1 1 1 2 2 3 3 3 4 4 ...
$ idreviewer: int 11 15 20 1 10 11 15 20 13 16 ...
$ score : num 7 7 7 7 7 7 7 7 7 7 ...
```

- The dataset `data.immer10` contains standard setting ratings of 13 raters on 61 items (including item identifier `item` and item difficulty `itemdiff`)

```
'data.frame': 61 obs. of 15 variables:
$ item : chr "I01" "I02" "I03" "I04" ...
$ itemdiff: num 380 388 397 400 416 425 427 434 446 459 ...
$ R01 : int 1 3 2 2 1 3 2 2 3 1 ...
$ R02 : int 1 1 1 1 1 2 1 2 2 1 ...
$ R03 : int 1 1 1 1 1 1 2 2 3 1 ...
$ R04 : int 1 2 1 3 2 2 2 2 3 2 ...
$ R05 : int 1 1 2 1 1 1 2 2 3 2 ...
$ R06 : int 1 2 1 1 1 2 2 2 3 2 ...
$ R07 : int 1 2 1 2 1 1 2 1 3 1 ...
$ R08 : int 2 2 1 2 1 1 2 2 3 2 ...
$ R09 : int 2 1 1 2 1 2 1 2 3 1 ...
$ R10 : int 2 2 2 2 1 2 2 3 3 2 ...
$ R11 : int 2 2 1 2 1 2 2 2 3 2 ...
$ R12 : int 2 2 1 3 1 2 2 2 3 2 ...
$ R13 : int 1 1 1 1 1 1 1 1 2 1 ...
```

- The dataset `data.immer11` contains ratings of 148 cases (screening mammogram samples) diagnoses by 110 raters (Zhang & Petersen, xxxx). The codes of the polytomous rating are normal (code 0), benign (code 1), probably benign (code 2), possibly malignant (code 3), and probably malignant (code 4). The dataset was extracted from an image plot in Figure 2 by using the processing function `png::readPNG`. The format of the dataset is

```
'data.frame': 148 obs. of 110 variables:
$ R001: num 2 1 3 2 1 2 0 0 0 2 ...
$ R002: num 1 3 4 4 0 4 0 0 3 0 ...
$ R003: num 0 0 0 4 0 2 3 0 0 0 ...
$ R004: num 1 2 1 4 2 2 2 0 4 4 ...
[... ]
```

- The dataset `data.immer12` contains ratings of the 2002 olympic pairs figure skating competition. This dataset has been used in Lincare (2009). The items are `ST` (short program, technical merit), `SA` (short program, artistic impression), `FT` (free program, technical merit), and `FA` (free program, artistic impression). The format of the dataset is

```
'data.frame': 180 obs. of 7 variables:
$ idpair: int 1 1 1 1 1 1 1 1 1 2 ...
$ pair : chr "BB-Svk" "BB-Svk" "BB-Svk" "BB-Svk" ...
$ judge : chr "RUS" "CHI" "USA" "FRA" ...
$ ST : int 58 57 57 56 55 55 50 51 51 47 ...
$ SA : int 58 57 57 56 55 55 50 51 51 47 ...
$ FT : int 58 57 57 56 55 55 50 51 51 47 ...
```

```
$ FA : int 58 57 57 56 55 55 50 51 51 47 ...
```

References

Kuhlisch, W., Roos, M., Rothe, J., Rudolph, J., Scheuermann, B., & Stoyan, D. (2016). A statistical approach to calibrating the scores of biased reviewers of scientific papers. *Metrika, 79*, 37-57.

Linacre, J. M. (2009). Local independence and residual covariance: A study of Olympic figure skating ratings. *Journal of Applied Measurement, 10*(2), 157-169.

Schuster, C., & Smith, D. A. (2006). Estimating with a latent class model the reliability of nominal judgments upon which two raters agree. *Educational and Psychological Measurement, 66*(5), 739-747.

Zhang, S., & Petersen, J. H. (XXXX). Quantifying rater variation for ordinal data using a rating scale model. *Statistics in Medicine, XX*(xx), xxx-xxx.

`data.ptam` *Example Datasets for Robitzsch and Steinfeld (2018)*

Description

Example datasets for Robitzsch and Steinfeld (2018).

Usage

```
data(data.ptam1)
data(data.ptam2)
data(data.ptam3)
data(data.ptam4)
data(data.ptam4long)
data(data.ptam4wide)
```

Format

- The dataset `data.ptam1` is a subset of the dataset from Example 3 of the ConQuest manual and contains 9395 ratings for 6877 students and 9 raters on 2 items (`OP` and `TF`). The format is

  ```
  'data.frame': 9395 obs. of 4 variables:
  $ pid : int 1508 1564 1565 1566 1567 1568 1569 1629 1630 1631 ...
  $ rater: num 174 124 124 124 124 124 124 114 114 114 ...
  $ OP : int 2 1 2 1 1 1 2 2 2 3 ...
  $ TF : int 3 1 2 2 1 1 2 2 2 3 ...
  ```

- The dataset `data.ptam2` contains 1043 ratings for 262 students and 17 raters on 19 items (`A1`, ..., `D9`). The format is

  ```
  'data.frame': 1043 obs. of 21 variables:
  $ idstud : int 1001 1001 1001 1001 1002 1002 1002 1002 1003 1003 ...
  $ idrater: int 101 108 212 215 104 108 209 211 103 104 ...
  ```

```
$ A1 : int 1 1 1 1 1 1 1 1 1 1 ...
$ A2 : int 1 1 1 1 0 0 0 1 1 1 ...
$ A3 : int 1 1 1 1 1 1 0 1 0 0 ...
[...]
$ D9 : int 2 2 2 2 2 2 2 2 1 0 ...
```

- The dataset `data.ptam3` contains 523 ratings for 262 students and 8 raters on 23 items (`A1`, ..., `J0`). The format is

```
'data.frame': 523 obs. of 25 variables:
$ idstud : int 1001 1001 1002 1002 1003 1003 1004 1004 1005 1005 ...
$ idrater: int 101 108 104 108 103 104 102 104 102 108 ...
$ A1 : int 1 1 1 1 1 1 1 1 1 1 ...
$ A2 : int 1 1 0 0 1 1 NA 0 1 1 ...
$ A3 : int 1 1 1 1 0 0 0 0 0 0 ...
[...]
$ J0 : int 2 3 3 2 0 0 2 2 0 1 ...
```

- The dataset `data.ptam4` contains 592 ratings for 209 students and 10 raters on 3 items (`crit2`, `crit3` and `crit4`). The format is

```
'data.frame': 592 obs. of 5 variables:
$ idstud: num 10005 10009 10010 10010 10014 ...
$ rater : num 802 802 844 802 837 824 820 803 816 844 ...
$ crit2 : int 3 2 0 2 1 0 2 1 1 0 ...
$ crit3 : int 3 2 1 2 2 2 2 2 2 2 ...
$ crit4 : int 2 1 2 1 2 2 2 2 2 2 ...
```

- The dataset `data.ptam4long` is the dataset `data.ptam4` which has been converted into a long format for analysis with mixed effects models in the **lme4** package. The format is

```
'data.frame': 1776 obs. of 17 variables:
$ idstud : num 10005 10005 10005 10009 10009 ...
$ rater : num 802 802 802 802 802 802 844 802 844 802 ...
$ item : Factor w/ 3 levels "crit2","crit3",..: 1 2 3 1 2 3 1 1 2 2 ...
$ value : int 3 3 2 2 2 1 0 2 1 2 ...
$ I_crit2: num 1 0 0 1 0 0 1 1 0 0 ...
$ I_crit3: num 0 1 0 0 1 0 0 0 1 1 ...
$ I_crit4: num 0 0 1 0 0 1 0 0 0 0 ...
$ R_802 : num 1 1 1 1 1 1 0 1 0 1 ...
$ R_803 : num 0 0 0 0 0 0 0 0 0 0 ...
[...]
$ R_844 : num 0 0 0 0 0 0 1 0 1 0 ...
```

- The dataset `data.ptam4wide` contains multiple ratings of 40 students from the dataset `data.ptam4` from the item `crit2`. Each column corresponds to one rater. The format is

```
'data.frame': 40 obs. of 11 variables:
$ pid : chr "10014" "10085" "10097" "10186" ...
$ R802: int 2 3 2 2 2 1 1 2 2 2 ...
$ R803: int 1 1 3 1 2 0 0 0 1 0 ...
```

```
$ R810: int 1 2 2 2 1 0 1 1 2 1 ...
$ R816: int 1 2 3 2 2 0 1 1 2 1 ...
$ R820: int 2 2 2 2 1 1 1 1 1 1 ...
$ R824: int 0 3 2 3 2 0 0 1 2 1 ...
$ R831: int 1 2 2 2 1 0 0 0 1 1 ...
$ R835: int 0 1 2 2 1 1 0 0 2 1 ...
$ R837: int 1 2 3 2 2 0 1 1 2 2 ...
$ R844: int 0 2 3 2 2 0 0 0 1 3 ...
```

References

Robitzsch, A., & Steinfeld, J. (2018). Item response models for human ratings: Overview, estimation methods, and implementation in R. *Psychological Test and Assessment Modeling, 60*(1), 101-139.

---

immer_agree2 *Agreement Statistics for 2 Raters*

Description

Some agreement statistics for two raters, including raw agreement, Scott's Pi, Cohen's Kappa, Gwets AC1 and Aickens Alpha (see Gwet, 2010).

Usage

```
immer_agree2(y, w=rep(1, nrow(y)), symmetrize=FALSE, tol=c(0, 1))

## S3 method for class 'immer_agree2'
summary(object, digits=3,...)
```

Arguments

y	Data frame with responses for two raters
w	Optional vector of frequency weights
symmetrize	Logical indicating whether contingency table should be symmetrized
tol	Vector of integers indicating tolerance for raw agreement
object	Object of class immer_agree2
digits	Number of digits after decimal for rounding
...	Further arguments to be passed

Value

List with entries

`agree_raw`	Raw agreement
`agree_stats`	Agreement statistics
`agree_table`	Contingency table
`marg`	Marginal frequencies
`Pe`	Expected chance agreement probabilities
`PH`	Probabilities for hard-to-classify subjects according to Aicken
`nobs`	Number of observations

References

Gwet, K. L. (2010). *Handbook of inter-rater reliability*. Gaithersburg: Advanced Analytics.

See Also

For more inter-rater agreement statistics see the R packages **agRee**, **Agreement**, **agrmt**, **irr**, **obs.agree**, **rel**.

Examples

```
#############################################################################
# EXAMPLE 1: Dataset in Schuster & Smith (2006)
#############################################################################

data(data.immer08)
dat <- data.immer08

y <- dat[,1:2]
w <- dat[,3]
# agreement statistics
res <- immer::immer_agree2( y=y, w=w )
summary(res)
# extract some output values
res$agree_stats
res$agree_raw
```

`immer_ccml` *Composite Conditional Maximum Likelihood Estimation for the Partial Credit Model with a Design Matrix for Item Parameters*

Description

Estimates the partial credit model with a design matrix for item parameters with composite conditional maximum likelihood estimation. The estimation uses pairs of items X_i and X_j and considers conditional likelihoods $P(X_i = k, X_j = h|\theta)/P(X_i + X_j = k + h|\theta)$. By using this strategy, the trait θ cancels out (like in conditional maximum likelihood estimation). The proposed strategy is a generalization of the Zwinderman (1995) composite conditional maximum likelihood approach of the Rasch model to the partial credit model. See Varin, Reid and Firth (2011) for a general introduction to composite conditional maximum likelihood estimation.

Usage

```
immer_ccml( dat, weights=NULL, irtmodel="PCM", A=NULL, b_fixed=NULL, control=NULL )

## S3 method for class 'immer_ccml'
summary(object, digits=3, file=NULL, ...)

## S3 method for class 'immer_ccml'
coef(object, ...)

## S3 method for class 'immer_ccml'
vcov(object, ...)
```

Arguments

`dat`	Data frame with polytomous item responses $0, 1, \ldots, K$
`weights`	Optional vector of sampling weights
`irtmodel`	Model string for specifying the item response model
`A`	Design matrix (items $\times$ categories $\times$ basis parameters). Entries for categories are for $1, \ldots, K$
`b_fixed`	Matrix with fixed b parameters
`control`	Control arguments for optimization function `stats::nlminb`
`object`	Object of class `immer_ccml`
`digits`	Number of digits after decimal to print
`file`	Name of a file in which the output should be sunk
`...`	Further arguments to be passed.

Details

The function estimates the partial credit model as $P(X_i = h|\theta) \propto \exp(h\theta - b_{ih})$ with $b_{ih} = \sum_l a_{ihl}\xi_l$ where the values a_{ihl} are included in the design matrix `A` and ξ_l denotes basis item parameters.

Value

List with following entries (selection)

coef	Item parameters
vcov	Covariance matrix for item parameters
se	Standard errors for item parameters
nlminb_result	Output from optimization with stats::nlminb
suff_stat	Used sufficient statistics
ic	Information criteria

References

Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica, 21*, 5-42.

Zwinderman, A. H. (1995). Pairwise parameter estimation in Rasch models. *Applied Psychological Measurement, 19*(4), 369-375.

See Also

See sirt::rasch.pairwise.itemcluster of an implementation of the composite conditional maximum likelihood approach for the Rasch model.

Examples

```
#############################################################################
# EXAMPLE 1: Partial credit model with CCML estimation
#############################################################################

library(TAM)

data(data.gpcm, package="TAM")
dat <- data.gpcm

#-- initial MML estimation in TAM to create a design matrix
mod1a <- TAM::tam.mml(dat, irtmodel="PCM2")
summary(mod1a)

#* define design matrix
A <- - mod1a$A[,-1,-1]
A <- A[,,-1]
str(A)

#-- estimate model
mod1b <- immer::immer_ccml( dat, A=A)
summary(mod1b)
```

`immer_cml` *Conditional Maximum Likelihood Estimation for the Linear Logistic Partial Credit Model*

Description

Conditional maximum likelihood estimation for the linear logistic partial credit model (Molenaar, 1995; Andersen, 1995; Fischer, 1995). The `immer_cml` function allows for known integer discrimination parameters like in the one-parameter logistic model (Verhelst & Glas, 1995).

Usage

```
immer_cml(dat, weights=NULL, W=NULL, b_const=NULL, par_init=NULL,
    a=NULL, irtmodel=NULL, normalization="first", nullcats="zeroprob",
    diff=FALSE, use_rcpp=FALSE, ...)

## S3 method for class 'immer_cml'
summary(object, digits=3, file=NULL, ...)

## S3 method for class 'immer_cml'
logLik(object,...)

## S3 method for class 'immer_cml'
anova(object,...)

## S3 method for class 'immer_cml'
coef(object,...)

## S3 method for class 'immer_cml'
vcov(object,...)
```

Arguments

`dat`	Data frame with item responses
`weights`	Optional vector of sample weights
`W`	Design matrix $\boldsymbol{W}$ for linear logistic partial credit model. Every row corresponds to a parameter for item i in category h
`b_const`	Optional vector of parameter constants b_{0ih} which can be used for parameter fixings.
`par_init`	Optional vector of initial parameter estimates
`a`	Optional vector of integer item discriminations
`irtmodel`	Type of item response model. `irtmodel="PCM"` and `irtmodel="PCM2"` follow the conventions of the **TAM** package.
`normalization`	The type of normalization in partial credit models. Can be `"first"` for the first item or `"sum"` for a sum constraint.

nullcats	A string indicating whether categories with zero frequencies should have a probability of zero (by fixing the constant parameter to a large value of 99).
diff	Logical indicating whether the difference algorithm should be used. See psychotools::elementary_sym for details.
use_rcpp	Logical indicating whether **Rcpp** package should be used for computation.
...	Further arguments to be passed to stats::optim.
object	Object of class immer_cml
digits	Number of digits after decimal to be rounded.
file	Name of a file in which the output should be sunk

Details

The partial credit model can be written as

$$P(X_{pi} = h) \propto \exp(a_i h \theta_p - b_{ih})$$

where the item-category parameters b_{ih} are linearly decomposed according to

$$b_{ih} = \sum_v w_{ihv} \beta_v + b_{0ih}$$

with unknown basis parameters β_v and fixed values w_{ihv} of the design matrix $\boldsymbol{W}$ (specified in W) and constants b_{0ih} (specified in b_const).

Value

List with following entries:

item	Data frame with item-category parameters
b	Item-category parameters b_{ih}
coefficients	Estimated basis parameters β_v
vcov	Covariance matrix of basis parameters β_v
par_summary	Summary for basis parameters
loglike	Value of conditional log-likelihood
deviance	Deviance
result_optim	Result from optimization in stats::optim
W	Used design matrix $\boldsymbol{W}$
b_const	Used constant vector b_{0ih}
par_init	Used initial parameters
suffstat	Sufficient statistics
score_freq	Score frequencies
dat	Used dataset
used_persons	Used persons
NP	Number of missing data patterns

N	Number of persons
I	Number of items
maxK	Maximum number of categories per item
K	Maximum score of all items
npars	Number of estimated parameters
pars_info	Information of definition of item-category parameters b_{ih}
parm_index	Parameter indices
item_index	Item indices
score	Raw score for each person

References

Andersen, E. B. (1995). Polytomous Rasch models and their estimation. In G. H. Fischer & I. W. Molenaar (Eds.). *Rasch Models* (pp. 39–52). New York: Springer.

Fischer, G. H. (1995). The linear logistic test model. In G. H. Fischer & I. W. Molenaar (Eds.). *Rasch Models* (pp. 131–156). New York: Springer.

Molenaar, I. W. (1995). Estimation of item parameters. In G. H. Fischer & I. W. Molenaar (Eds.). *Rasch Models* (pp. 39–52). New York: Springer.

Verhelst, N. D. &, Glas, C. A. W. (1995). The one-parameter logistic model. In G. H. Fischer & I. W. Molenaar (Eds.). *Rasch Models* (pp. 215–238). New York: Springer.

See Also

For CML estimation see also the **eRm** and **psychotools** packages and the functions `eRm::RM` and `psychotools::raschmodel` for the Rasch model and `eRm::PCM` and `psychotools::pcmodel` for the partial credit model.

See `eRm::LLTM` for the linear logistic test model and `eRm::LPCM` for the linear logistic partial credit model in the **eRm** package for CML implementations.

The `immer_cml` function makes use of `psychotools::elementary_symmetric_functions`.

For CML estimation with sample weights see also the **RM.weights** package.

Examples

```
#############################################################################
# EXAMPLE 1: Dichotomous data data.read
#############################################################################

library(sirt)
library(psychotools)
library(TAM)
library(CDM)

data(data.read, package="sirt")
dat <- data.read
I <- ncol(dat)
```

```
#---------------------------------------------------------------
#--- Model 1: Rasch model, setting first item difficulty to zero
mod1a <- immer::immer_cml( dat=dat)
summary(mod1a)
logLik(mod1a) # extract log likelihood
coef(mod1a)   # extract coefficients

## Not run:
library(eRm)

# estimate model in psychotools package
mod1b <- psychotools::raschmodel(dat)
summary(mod1b)
logLik(mod1b)

# estimate model in eRm package
mod1c <- eRm::RM(dat, sum0=FALSE)
summary(mod1c)
mod1c$etapar

# compare estimates of three packages
cbind( coef(mod1a), coef(mod1b), mod1c$etapar )

#---------------------------------------------------------------
#-- Model 2: Rasch model sum normalization
mod2a <- immer::immer_cml( dat=dat, normalization="sum")
summary(mod2a)

# compare estimation in TAM
mod2b <- tam.mml( dat, constraint="items"  )
summary(mod2b)
mod2b$A[,2,]

#---------------------------------------------------------------
#--- Model 3: some fixed item parameters
# fix item difficulties of items 1,4,8
# define fixed parameters in constant parameter vector
b_const <- rep(0,I)
fix_items <- c(1,4,8)
b_const[ fix_items ] <- c( -2.1, .195, -.95 )
# design matrix
W <- matrix( 0, nrow=12, ncol=9)
W[ cbind( setdiff( 1:12, fix_items ), 1:9 ) ] <- 1
colnames(W) <- colnames(dat)[ - fix_items ]
# estimate model
mod3 <- immer::immer_cml( dat=dat, W=W, b_const=b_const)
summary(mod3)

#---------------------------------------------------------------
#--- Model 4: One parameter logistic model
# estimate non-integer item discriminations with 2PL model
I <- ncol(dat)
mod4a <- sirt::rasch.mml2( dat, est.a=1:I )
```

```
summary(mod4a)
a <- mod4a$item$a     # extract (non-integer) item discriminations
# estimate integer item discriminations ranging from 1 to 3
a_integer <- immer::immer_opcat( a, hmean=2, min=1, max=3 )
# estimate one-parameter model with fixed integer item discriminations
mod4 <- immer::immer_cml( dat=dat, a=a_integer )
summary(mod4)

#-----------------------------------------------------------------
#--- Model 5: Linear logistic test model

# define design matrix
W <- matrix( 0, nrow=12, ncol=5 )
colnames(W) <- c("B","C", paste0("Pos", 2:4))
rownames(W) <- colnames(dat)
W[ 5:8, "B" ] <- 1
W[ 9:12, "C" ] <- 1
W[ c(2,6,10), "Pos2" ] <- 1
W[ c(3,7,11), "Pos3" ] <- 1
W[ c(4,8,12), "Pos4" ] <- 1

# estimation with immer_cml
mod5a <- immer::immer_cml( dat, W=W )
summary(mod5a)

# estimation in eRm package
mod5b <- eRm::LLTM( dat, W=W )
summary(mod5b)

# compare models 1 and 5 by a likelihood ratio test
anova( mod1a, mod5a )

#############################################################################
# EXAMPLE 2: Polytomous data | data.Students
#############################################################################

data(data.Students,package="CDM")
dat <- data.Students
dat <- dat[, grep("act", colnames(dat) ) ]
dat <- dat[1:400,]  # select a subdataset
dat <- dat[ rowSums( 1 - is.na(dat) ) > 1, ]
    # remove persons with less than two valid responses

#-----------------------------------------------------------------
#--- Model 1: Partial credit model with constraint on first parameter
mod1a <- immer::immer_cml( dat=dat )
summary(mod1a)
# compare pcmodel function from psychotools package
mod1b <- psychotools::pcmodel( dat )
summary(mod1b)
# estimation in eRm package
mod1c <- eRm::PCM( dat, sum0=FALSE )
  # -> subjects with only one valid response must be removed
```

```
summary(mod1c)

#----------------------------------------------------------------
#-- Model 2: Partial credit model with sum constraint on item difficulties
mod2a <- immer::immer_cml( dat=dat, irtmodel="PCM2", normalization="sum")
summary(mod2a)
# compare with estimation in TAM
mod2b <- TAM::tam.mml( dat, irtmodel="PCM2", constraint="items")
summary(mod2b)

#----------------------------------------------------------------
#-- Model 3: Partial credit model with fixed integer item discriminations
mod3 <- immer::immer_cml( dat=dat, normalization="first", a=c(2,2,1,3,1) )
summary(mod3)

#############################################################################
# EXAMPLE 3: Polytomous data | Extracting the structure of W matrix
#############################################################################

data(data.mixed1, package="sirt")
dat <- data.mixed1

# use non-exported function "lpcm_data_prep" to extract the meaning
# of the rows in W which are contained in value "pars_info"
res <- immer:::lpcm_data_prep( dat, weights=NULL, a=NULL )
pi2 <- res$pars_info

# create design matrix with some restrictions on item parameters
W <- matrix( 0, nrow=nrow(pi2), ncol=2 )
colnames(W) <- c( "P2", "P3" )
rownames(W) <- res$parnames

# joint item parameter for items I19 and I20 fixed at zero
# item parameter items I21 and I22
W[ 3:10, 1 ] <- pi2$cat[ 3:10 ]
# item parameters I23, I24 and I25
W[ 11:13, 2] <- 1

# estimate model with design matrix W
mod <- immer::immer_cml( dat, W=W)
summary(mod)

#############################################################################
# EXAMPLE 4: Partial credit model with raters
#############################################################################

data(data.immer07)
dat <- data.immer07

#*** reshape dataset for one variable
dfr1 <- immer::immer_reshape_wideformat( dat$I1, rater=dat$rater, pid=dat$pid )

#-- extract structure of design matrix
```

```
res <- immer:::lpcm_data_prep( dat=dfr1[,-1], weights=NULL, a=NULL)
pars_info <- res$pars_info

# specify design matrix for partial credit model and main rater effects
# -> set sum of all rater effects to zero
W <- matrix( 0, nrow=nrow(pars_info), ncol=3+2 )
rownames(W) <- rownames(pars_info)
colnames(W) <- c( "Cat1", "Cat2", "Cat3", "R1", "R2" )
# define item parameters
W[ cbind( pars_info$index, pars_info$cat ) ] <- 1
# define rater parameters
W[ paste(pars_info$item)=="R1", "R1" ] <- 1
W[ paste(pars_info$item)=="R2", "R2" ] <- 1
W[ paste(pars_info$item)=="R3", c("R1","R2") ] <- -1
# set parameter of first category to zero for identification constraints
W <- W[,-1]

# estimate model
mod <- immer::immer_cml( dfr1[,-1], W=W)
summary(mod)

#############################################################################
# EXAMPLE 5: Multi-faceted Rasch model | Estimation with a design matrix
#############################################################################

data(data.immer07)
dat <- data.immer07

#*** reshape dataset
dfr1 <- immer::immer_reshape_wideformat( dat[, paste0("I",1:4) ], rater=dat$rater,
                pid=dat$pid )

#-- structure of design matrix
res <- immer:::lpcm_data_prep( dat=dfr1[,-1], weights=NULL, a=NULL)
pars_info <- res$pars_info

#--- define design matrix for multi-faceted Rasch model with only main effects
W <- matrix( 0, nrow=nrow(pars_info), ncol=3+2+2 )
parnames <- rownames(W) <- rownames(pars_info)
colnames(W) <- c( paste0("I",1:3), paste0("Cat",1:2), paste0("R",1:2) )
#+ define item effects
for (ii in c("I1","I2","I3") ){
    ind <- grep( ii, parnames )
    W[ ind, ii ] <- pars_info$cat[ind ]
                }
ind <- grep( "I4", parnames )
W[ ind, c("I1","I2","I3") ] <- -pars_info$cat[ind ]
#+ define step parameters
for (cc in 1:2 ){
    ind <- which( pars_info$cat==cc )
    W[ ind, paste0("Cat",1:cc) ] <- 1
                }
#+ define rater effects
```

```
for (ii in c("R1","R2") ){
    ind <- grep( ii, parnames )
    W[ ind, ii ] <- pars_info$cat[ind ]
                }
ind <- grep( "R3", parnames )
W[ ind, c("R1","R2") ] <- -pars_info$cat[ind ]

#--- estimate model with immer_cml
mod1 <- immer::immer_cml( dfr1[,-1], W=W, par_init=rep(0,ncol(W) ) )
summary(mod1)

#--- comparison with estimation in TAM
resp <- dfr1[,-1]
mod2 <- TAM::tam.mml.mfr( resp=dat[,-c(1:2)], facets=dat[, "rater", drop=FALSE ],
            pid=dat$pid, formulaA=~ item + step + rater )
summary(mod2)

## End(Not run)
```

`immer_FACETS` *Wrapper to FACDOS*

Description

This Function is a wrapper do the DOS version of FACETS (Linacre, 1999).

Usage

```
immer_FACETS(title=NULL, convergence=NULL, totalscore=NULL, facets=NULL,
  noncenter=NULL,  arrange=NULL,entered_in_data=NULL, models=NULL,
  inter_rater=NULL, pt_biserial=NULL, faire_score=NULL, unexpected=NULL,
  usort=NULL, positive=NULL, labels=NULL, fileinput=NULL, data=NULL,
  path.dosbox=NULL, path.facets="", model.name=NULL, facetsEXE=NULL )
```

Arguments

`title`	title of the analysis
`convergence`	convergence criteria
`totalscore`	show the total score with each observation
`facets`	number of specified facets
`noncenter`	specified the non centered facet here
`arrange`	control the ordering in each table/output
`entered_in_data`	optional specification for facets
`models`	model to be used in the analysis
`inter_rater`	Specify rater facet number for the agreement report among raters

`pt_biserial`	correlation between the raw-score for each element
`faire_score`	intended for communicating the measures as adjusted ratings
`unexpected`	size of smallest standardized residual
`usort`	order in which the unexpected observation are listed
`positive`	specifies which facet is positively oriented
`labels`	name of each facet, followed by a list of elements
`fileinput`	optional argument, if your data are stored within a separate file
`data`	Input of the data in long-format
`path.dosbox`	Path to the installed DOSBox. If `NULL`: the function assumed that you have purchased FACETS and would like to use this version (currently only for Windows-User)
`path.facets`	Path to FACDOS or FACETS if the path.dosbox is "NULL"
`model.name`	Name of the configuration file for FACETS
`facetsEXE`	optional argument to specifie specific FACETS.exe

Details

Within the function `immer_FACETS` it is either possible to pass existing FACETS input files or to specify the Input within the function. To run the estimation in FACETS it is necessary to provide both the path to the DosBox and FACDOS (it is recommended to use the function `immer_install` for the installation process). After the estimation process is finished the Exports are in the Facets folder.

References

Linacre, J. M. (1999). *FACETS* (Version 3.17)[Computer software]. Chicago: MESA.

See Also

Install FACDOS and DOSBox `immer_install`.

Examples

```
## Not run:
#################################
# 1. Example on Windows
#################################
# define data generating parameters
set.seed(1997)
N <- 500  # number of persons
I <- 4    # number of items
R <- 3    # number of raters
K <- 3    # maximum score
sigma <- 2  # standard deviation
theta <- rnorm( N, sd=sigma )  # abilities
# item intercepts
b <- outer( seq( -1.5, 1.5, len=I), seq( -2, 2, len=K), "+" )
```

```
# item loadings
a <- rep(1,I)
# rater severity parameters
phi <- matrix( c(-.3, -.2, .5), nrow=I, ncol=R, byrow=TRUE )
phi <- phi + rnorm( phi, sd=.3 )
phi <- phi - rowMeans(phi)
# rater variability parameters
psi <- matrix( c(.1, .4, .8), nrow=I, ncol=R, byrow=TRUE )
# simulate HRM data
data <- immer::immer_hrm_simulate( theta, a, b, phi=phi, psi=psi )

# prepare data for FACETS
data2FACETS <- function(data){
  tmp <- match(c("pid","rater"),colnames(data))
  items <- grep("I",colnames(data))
cbind(data[, match(c("pid","rater"),
colnames(data))],gr=paste0("1-",length(items)),data[,items])
}
facets_in <- data2FACETS(data)

# Example of FACETS
mod1.a <- immer::immer_FACETS(
  title="Example 1 with simulated data",
  convergence=NULL,
  totalscore="YES",
  facets=3,
  noncenter=NULL,
  arrange="m,N",
  entered_in_data="2,1,1",
  models="?$,?$,?$,R4",
  inter_rater=NULL,
  pt_biserial=NULL,
  faire_score="Zero",
  unexpected=2,
  usort=NULL,
  positive=1,
  labels=c("1,Persons","1-500","2,Rater","1-3","3,Item","1-4"),
  fileinput=NULL,
  data=facets_in,
  path.dosbox=NULL,
  path.facets="C:\\Facets",
  model.name="Example.SD",
  facetsEXE=NULL
  )

#################################
# 2. Example on Windows using existing input-files of FACETS
#################################
data(data.immer06)

mod1b <- immer::immer_FACETS(
  fileinput=data.immer06,
  path.facets="C:\\Facets",
```

```
  model.name="Example.SD",
  facetsEXE=NULL
  )

## End(Not run)
```

---

immer_hrm *Hierarchical Rater Model (Patz et al., 2002)*

Description

Estimates the hierarchical rater model (HRM; Patz et al., 2002; see Details) with Markov Chain Monte Carlo using a Metropolis-Hastings algorithm.

Usage

```
immer_hrm(dat, pid, rater, iter, burnin, N.save=3000, prior=NULL,  est.a=FALSE,
           est.sigma=TRUE,  est.mu=FALSE, est.phi="a", est.psi="a",
       MHprop=NULL, theta_like=seq(-10,10,len=30), sigma_init=1, print_iter=20 )

## S3 method for class 'immer_hrm'
summary(object, digits=3, file=NULL, ...)

## S3 method for class 'immer_hrm'
plot(x,...)

## S3 method for class 'immer_hrm'
logLik(object,...)

## S3 method for class 'immer_hrm'
anova(object,...)

## S3 method for class 'immer_hrm'
IRT.likelihood(object,...)

## S3 method for class 'immer_hrm'
IRT.posterior(object,...)
```

Arguments

dat	Data frame with item responses
pid	Person identifiers
rater	Rater identifiers
iter	Number of iterations
burnin	Number of burnin iterations
N.save	Maximum number of samples to be saved.

prior	Parameters for prior distributions
est.a	Logical indicating whether a parameter should be estimated.
est.sigma	Logical indicating whether σ parameter should be estimated.
est.mu	Optional logical indicating whether the mean μ of the trait θ should be estimated.
est.phi	Type of ϕ_{ir} parameters to be estimated. If `est.phi="a"`, then ϕ_{ir} is estimated for all items and all raters. If `est.phi="r"`, then $\phi_{ir} = \phi_r$ is rater specific, while for `est.phi="i"` it is item specific ($\phi_{ir} = \phi_i$). In case of `est.phi="n"`, no ϕ parameters are estimated and all ϕ parameters are fixed at 0.
est.psi	Type of ψ_{ir} parameters to be estimated. The arguments follow the same conventions as `est.phi`, but also allows `est.psi="e"` (exchangeable) which means $\psi_{ir} = \psi$, i.e assuming the same ψ parameter for all items and raters.
MHprop	Parameters for Metropolis Hastings sampling. The standard deviation of the proposal distribution is adaptively computed (Browne & Draper, 2000).
theta_like	Grid of θ values to be used for likelihood approximation
sigma_init	Initial value for `sigma`
print_iter	Integer indicating that after each `print_iter`th iteration output on the console should be displayed.
object	Object of class `immer_hrm`
digits	Number of digits after decimal to print
file	Name of a file in which the output should be sunk
x	Object of class `immer_hrm`
...	Further arguments to be passed. See `sirt::plot.mcmc.sirt` for options in `plot`.

Details

The hierarchical rater model is defined at the level of persons

$$P(\xi_{pi} = \xi | \theta_p) \propto \exp(\xi \cdot a_i \cdot \theta_p - b_{ik})$$

where θ_p is normally distributed with mean μ and standard deviation σ.

At the level of ratings, the model is defined as

$$P(X_{pir} = x | \theta_p, \xi_{pi}) \propto \exp(-(x - \xi_{pi} - \phi_{ir})^2 / (2 \cdot \psi_{ir}))$$

Value

A list with following entries

person	Data frame with estimated person parameters
item	Data frame with estimated item parameters
rater_pars	Data frame with estimated rater parameters
est_pars	Estimated item and trait distribution parameters arranged in vectors and matrices.

sigma	Estimated standard deviation σ of trait θ
mu	Estimated mean μ of trait θ
mcmcobj	Object of class mcmc.list for **coda** package.
summary.mcmcobj	Summary of all parameters
EAP.rel	EAP reliability
ic	Parameters for information criteria
f.yi.qk	Individual likelihood evaluated at theta_like
f.qk.yi	Individual posterior evaluated at theta_like
theta_like	Grid of θ values for likelihood approximation
pi.k	Discretized θ distribution
like	Log-likelihood value
MHprop	Updated parameters in Metropolis-Hastings sampling

References

Browne, W. J., & Draper, D. (2000). Implementation and performance issues in the Bayesian and likelihood fitting of multilevel models. *Computational Statistics, 15*, 391-420.

Patz, R. J., Junker, B. W., Johnson, M. S., & Mariano, L. T. (2002). The hierarchical rater model for rated test items and its application to large-scale educational assessment data. *Journal of Educational and Behavioral Statistics, 27*(4), 341-384.

See Also

Simulate the HRM with immer_hrm_simulate.

Examples

```
## Not run:
library(sirt)
library(TAM)

#############################################################################
# EXAMPLE 1: Simulated data using the immer::immer_hrm_simulate() function
#############################################################################

# define data generating parameters
set.seed(1997)
N <- 500  # number of persons
I <- 4    # number of items
R <- 3    # number of raters
K <- 3    # maximum score
sigma <- 2  # standard deviation
theta <- stats::rnorm( N, sd=sigma )  # abilities
# item intercepts
b <- outer( seq( -1.5, 1.5, len=I), seq( -2, 2, len=K), "+" )
# item loadings
```

```
a <- rep(1,I)
# rater severity parameters
phi <- matrix( c(-.3, -.2, .5), nrow=I, ncol=R, byrow=TRUE )
phi <- phi + stats::rnorm( phi, sd=.3 )
phi <- phi - rowMeans(phi)
# rater variability parameters
psi <- matrix( c(.1, .4, .8), nrow=I, ncol=R, byrow=TRUE )
# simulate HRM data
data <- immer::immer_hrm_simulate( theta, a, b, phi=phi, psi=psi )
pid <- data$pid
rater <- data$rater
dat <- data[, - c(1:2) ]

#----------------------------------------------------------------
#*** Model 1: estimate HRM with equal item slopes
iter <- 3000
burnin <- 500
mod1 <- immer::immer_hrm( dat, pid, rater, iter=iter,  burnin=burnin )
summary(mod1)
plot(mod1,layout=2,ask=TRUE)

# relations among convergence diagnostic statistics
par(mfrow=c(1,2))
plot( mod1$summary.mcmcobj$PercVarRatio, log(mod1$summary.mcmcobj$effSize), pch=16)
plot( mod1$summary.mcmcobj$PercVarRatio, mod1$summary.mcmcobj$Rhat, pch=16)
par(mfrow=c(1,1))

# extract individual likelihood
lmod1 <- IRT.likelihood(mod1)
str(lmod1)
# extract log-likelihood value
logLik(mod1)

# write coda files into working directory
sirt::mcmclist2coda(mod1$mcmcobj, name="hrm_mod1")

#----------------------------------------------------------------
#*** Model 2: estimate HRM with estimated item slopes
mod2 <- immer::immer_hrm( dat, pid, rater, iter=iter,  burnin=burnin,
            est.a=TRUE, est.sigma=FALSE)
summary(mod2)
plot(mod2,layout=2,ask=TRUE)

# model comparison
anova( mod1, mod2 )

#----------------------------------------------------------------
#*** Model 3: Some prior specifications
prior <- list()
# prior on mu
prior$mu$M <- .7
prior$mu$SD <- 5
# fixed item parameters for first item
```

```
prior$b$M <-  matrix( 0, nrow=4, ncol=3 )
prior$b$M[1,] <- c(-2,0,2)
prior$b$SD <-  matrix( 10, nrow=4, ncol=3 )
prior$b$SD[1,] <- 1E-4
# estimate model
mod3 <- immer::immer_hrm( dat, pid, rater, iter=iter,  burnin=burnin, prior=prior)
summary(mod3)
plot(mod3)

#-----------------------------------------------------------------
#*** Model 4: Multi-faceted Rasch models in TAM package

# create facets object
facets <- data.frame( "rater"=rater )

#-- Model 4a: only main rater effects
form <- ~ item*step + rater
mod4a <- TAM::tam.mml.mfr( dat, pid=pid, facets=facets, formulaA=form)
summary(mod4a)

#-- Model 4b: item specific rater effects
form <- ~ item*step + item*rater
mod4b <- TAM::tam.mml.mfr( dat, pid=pid, facets=facets, formulaA=form)
summary(mod4b)

#-----------------------------------------------------------------
#*** Model 5: Faceted Rasch models with sirt::rm.facets

#--- Model 5a: Faceted Rasch model with only main rater effects
mod5a <- sirt::rm.facets( dat, pid=pid, rater=rater )
summary(mod5a)

#--- Model 5b: allow rater slopes for different rater discriminations
mod5b <- sirt::rm.facets( dat, pid=pid, rater=rater, est.a.rater=TRUE )
summary(mod5b)

#############################################################################
# EXAMPLE 2: data.ratings1 (sirt package)
#############################################################################

data(data.ratings1, package="sirt")
dat <- data.ratings1

# set number of iterations and burnin iterations
set.seed(87)
iter <- 1000
burnin <- 500
# estimate model
mod <- immer::immer_hrm( dat[,  paste0("k",1:5) ], pid=dat$idstud, rater=dat$rater,
             iter=iter, burnin=burnin )
summary(mod)
plot(mod, layout=1, ask=TRUE)
plot(mod, layout=2, ask=TRUE)
```

```
## End(Not run)
```

`immer_hrm_simulate` *Simulating the Hierarchical Rater Model (Patz et al., 2002)*

Description

Simulates the hierarchical rater model (Patz et al., 2002).

Usage

```
immer_hrm_simulate(theta, a, b, phi, psi)
```

Arguments

`theta`	Vector of θ parameters
`a`	Vector of a parameters
`b`	Matrix of b parameters
`phi`	Matrix of ϕ parameters
`psi`	Matrix of ψ parameters

Details

See `immer_hrm` for more details of the hierarchical rater model.

Value

Dataset with simulated item responses as well as vectors of person and rater identifiers

References

Patz, R. J., Junker, B. W., Johnson, M. S., & Mariano, L. T. (2002). The hierarchical rater model for rated test items and its application to large-scale educational assessment data. *Journal of Educational and Behavioral Statistics, 27*(4), 341-384.

See Also

See Example 1 in `immer_hrm` for applying the `immer_hrm_simulate` function.

immer_install — *Support for the installation of the DOS-version from FACETS*

Description

This function supports the installation process of the DOS-version from FACETS and also the necessary DOSBox in Windows, Linux (Ubuntu) and OS X

Usage

```
immer_install(DosBox_path=NULL, Facets_path=NULL )
```

Arguments

`DosBox_path`	optional argument for the specification of the path where the DosBox should be saved
`Facets_path`	optional argument for the specification of the path where FACETS should be saved

Details

This function provides assistance for the installation process of the FACDOS (DOS version of FACETS) and the required DosBox. Currently supported operating systems are: Windows, Mac OS X and Ubuntu (Linux).

References

Linacre, J. M. (1999). *FACETS* (Version 3.17) [Computer software]. Chicago: MESA.

Veenstra, P., Froessman, T., Wohlers, U. (2015): *DOSBox* (Version 0.74) [Computer Software]. Arizona: Scottsdale.

See Also

Install FACDOS and DOSBox `immer_FACETS`.

Examples

```
## Not run:
    immer::immer_install( DosBox_path=NULL, Facets_path=NULL )

## End(Not run)
```

---

immer_jml *Joint Maximum Likelihood Estimation for the Partial Credit Model with a Design Matrix for Item Parameters and ε-Adjustment Bias Correction*

Description

Estimates the partial credit model with a design matrix for item parameters with joint maximum likelihood (JML). The ε-adjustment bias correction is implemented with reduces bias of the JML estimation method (Bertoli-Barsotti, Lando & Punzo, 2014).

Usage

```
immer_jml(dat, A=NULL, maxK=NULL, center_theta=TRUE, b_fixed=NULL, weights=NULL,
   irtmodel="PCM", pid=NULL, rater=NULL, eps=0.3, est_method="eps_adj", maxiter=1000,
   conv=1e-05, max_incr=3, incr_fac=1.1, maxiter_update=10, maxiter_line_search=6,
     conv_update=1e-05, verbose=TRUE, use_Rcpp=TRUE, shortcut=TRUE)

## S3 method for class 'immer_jml'
summary(object, digits=3, file=NULL, ...)

## S3 method for class 'immer_jml'
logLik(object, ...)

## S3 method for class 'immer_jml'
IRT.likelihood(object, theta=seq(-9,9,len=41), ...)
```

Arguments

`dat`	Data frame with polytomous item responses $0, 1, \ldots, K$
`A`	Design matrix (items $\times$ categories $\times$ basis parameters). Entries for categories are for $1, \ldots, K$
`maxK`	Optional vector with maximum category per item
`center_theta`	Logical indicating whether the trait estimates should be centered
`b_fixed`	Matrix with fixed b parameters
`irtmodel`	Specified item response model. Can be one of the two partial credit model parametrizations `PCM` and `PCM2`.
`weights`	Optional vector of sampling weights
`pid`	Person identifier
`rater`	Optional rater identifier
`eps`	Adjustment parameter ε
`est_method`	Estimation method. Can be `'eps_adj'` for the ε-adjustment, `'jml'` for the JML without bias correction and `'jml_bc'` for JML with bias correction.
`maxiter`	Maximum number of iterations

conv	Convergence criterion
max_incr	Maximum increment
incr_fac	Factor for shrinking increments from max_incr in every iteration
maxiter_update	Maximum number of iterations for parameter updates
maxiter_line_search	Maximum number of iterations within line search
conv_update	Convergence criterion for updates
verbose	Logical indicating whether convergence progress should be displayed
use_Rcpp	Logical indicating whether **Rcpp** package should be used for computation.
shortcut	Logical indicating whether a computational shortcut should be used for efficiency reasons
object	Object of class immer_jml
digits	Number of digits after decimal to print
file	Name of a file in which the output should be sunk
theta	Grid of θ values
...	Further arguments to be passed.

Details

The function uses the partial credit model as $P(X_i = h|\theta) \propto \exp(h\theta - b_{ih})$ with $b_{ih} = \sum_l a_{ihl}\xi_l$ where the values a_{ihl} are included in the design matrix A and ξ_l denotes basis item parameters.

The adjustment parameter ε is applied to the sum score as the sufficient statistic for the person parameter. In more detail, extreme scores $S_p = 0$ (minimum score) or $S_p = M_p$ (maximum score) are adjusted to $S_p^* = \varepsilon$ or $S_p^* = M_p - \varepsilon$, respectively. Therefore, the adjustment possesses more influence on parameter estimation for datasets with a small number of items.

Value

List with following entries

b	Item parameters b_{ih}
theta	Person parameters
theta_se	Standard errors for person parameters
xsi	Basis parameters
xsi_se	Standard errors for bias parameters
probs	Predicted item response probabilities
person	Data frame with person scores
dat_score	Scoring matrix
score_pers	Sufficient statistics for persons
score_items	Sufficient statistics for items
loglike	Log-likelihood value

References

Bertoli-Barsotti, L., Lando, T., & Punzo, A. (2014). Estimating a Rasch Model via fuzzy empirical probability functions. In D. Vicari, A. Okada, G. Ragozini & C. Weihs (Eds.). *Analysis and Modeling of Complex Data in Behavioral and Social Sciences*, Springer.

See Also

See `TAM::tam.jml` for joint maximum likelihood estimation. The $varepsilon$-adjustment is also implemented in `sirt::mle.pcm.group`.

Examples

```
#############################################################################
# EXAMPLE 1: Rasch model
#############################################################################

data(data.read, package="sirt")
dat <- data.read

#---  Model 1: Rasch model with JML and epsilon-adjustment
mod1a <- immer::immer_jml(dat)
summary(mod1a)

## Not run:
#- JML estimation, only handling extreme scores
mod1b <- immer::immer_jml( dat, est_method="jml")
summary(mod1b)

#- JML estimation with (I-1)/I bias correction
mod1c <- immer::immer_jml( dat, est_method="jml_bc" )
summary(mod1c)

# compare different estimators
round( cbind( eps=mod1a$xsi, JML=mod1b$xsi, BC=mod1c$xsi ), 2 )

#---  Model 2: LLTM by defining a design matrix for item difficulties
A <- array(0, dim=c(12,1,3) )
A[1:4,1,1] <- 1
A[5:8,1,2] <- 1
A[9:12,1,3] <- 1

mod2 <- immer::immer_jml(dat, A=A)
summary(mod2)

#############################################################################
# EXAMPLE 2: Partial credit model
#############################################################################

library(TAM)
data(data.gpcm, package="TAM")
dat <- data.gpcm
```

```
#-- JML estimation in TAM
mod0 <- TAM::tam.jml(resp=dat, bias=FALSE)
summary(mod0)

# extract design matrix
A <- mod0$A
A <- A[,-1,]

#-- JML estimation
mod1 <- immer::immer_jml(dat, A=A, est_method="jml")
summary(mod1)

#-- JML estimation with epsilon-adjusted bias correction
mod2 <- immer::immer_jml(dat, A=A, est_method="eps_adj")
summary(mod2)

#############################################################################
# EXAMPLE 3: Rating scale model with raters | Use design matrix from TAM
#############################################################################

data(data.ratings1, package="sirt")
dat <- data.ratings1

facets <- dat[,"rater", drop=FALSE]
resp <- dat[,paste0("k",1:5)]

#* Model 1: Rating scale model in TAM
formulaA <- ~ item + rater + step
mod1 <- TAM::tam.mml.mfr(resp=resp, facets=facets, formulaA=formulaA,
                pid=dat$idstud)
summary(mod1)

#* Model 2: Same model estimated with JML
resp0 <- mod1$resp
A0 <- mod1$A[,-1,]
mod2 <- immer::immer_jml(dat=resp0, A=A0, est_method="eps_adj")
summary(mod2)

## End(Not run)
```

---

immer_latent_regression *Unidimensional Latent Regression*

Description

Fits a unidimensional latent regression $\theta_{ig} = Y_{ig}\boldsymbol{\beta} + \varepsilon_{ig}$ with group-specific variances $Var(\varepsilon_{ig}) = \sigma_g^2$ based on the individual likelihood of a fitted model.

Usage

```
immer_latent_regression(like, theta=NULL, Y=NULL, group=NULL, weights=NULL,
   conv=1e-05, maxit=200, verbose=TRUE)

## S3 method for class 'immer_latent_regression'
summary(object, digits=3, file=NULL, ...)

## S3 method for class 'immer_latent_regression'
coef(object, ...)

## S3 method for class 'immer_latent_regression'
vcov(object, ...)

## S3 method for class 'immer_latent_regression'
logLik(object, ...)

## S3 method for class 'immer_latent_regression'
anova(object, ...)
```

Arguments

`like`	Matrix containing the individual likelihood $L(\boldsymbol{X}\vert\theta)$
`theta`	Grid of $\boldsymbol{\theta}$ values
`Y`	Predictor matrix
`group`	Group identifiers
`weights`	Optional vector of weights
`conv`	Convergence criterion
`maxit`	Maximum number of iterations
`verbose`	Logical indicating whether progress should be displayed
`object`	Object of class `immer_latent_regression`
`digits`	Number of digits after decimal to print
`file`	Name of a file in which the output should be sunk
`...`	Further arguments to be passed.

Value

List containing values (selection)

`coef`	Parameter vector
`vcov`	Covariance matrix for estimated parameters
`beta`	Regression coefficients
`gamma`	Standard deviations
`beta_stat`	Data frame with $\boldsymbol{\beta}$ parameters
`gamma_stat`	Data frame with standard deviations

`ic`	Information criteria
`deviance`	Deviance
`N`	Number of persons
`G`	Number of groups
`group`	Group identifier
`iter`	Number of iterations

Note

The `IRT.likelihood` method can be used for extracting the individual likelihood.

References

Adams, R. J., & Wu, M. L. (2007). The mixed-coefficients multinomial logit model. A generalized form of the Rasch model. In M. von Davier & C. H. Carstensen (Eds.): *Multivariate and mixture distribution Rasch models: Extensions and applications* (pp. 55-76). New York: Springer.

See Also

See `TAM::tam.latreg` for latent regression estimation in the **TAM** package.

Examples

```
## Not run:
#############################################################################
# EXAMPLE 1: Latent regression for Rasch model with simulated data
#############################################################################

library(sirt)

#-- simulate data
set.seed(9877)
I <- 15  # number of items
N <- 700 # number of persons per group
G <- 3   # number of groups
b <- seq(-2,2,len=I)
group <- rep( 1:G, each=N)
mu <- seq(0,1, length=G)
sigma <- seq(1, 1.5, length=G)
dat <- sirt::sim.raschtype( stats::rnorm( N*G, mean=mu[group], sd=sigma[group] ), b)

#-- estimate Rasch model with JML
mod1 <- immer::immer_jml( dat )
summary(mod1)

#-- compute individual likelihood
like1 <- IRT.likelihood(mod1)

#-- estimate latent regression
mod2 <- immer::immer_latent_regression( like=like1, group=group)
```

```
summary(mod2)

## End(Not run)
```

---

immer_opcat *Estimation of Integer Item Discriminations*

Description

Estimates integer item discriminations like in the one-parameter logistic model (OPLM; Verhelst & Glas, 1995). See Verhelst, Verstralen and Eggen (1991) for computational details.

Usage

```
immer_opcat(a, hmean, min=1, max=10, maxiter=200)
```

Arguments

a	Vector of estimated item discriminations
hmean	Prespecified harmonic mean
min	Minimum integer item discrimination
max	Maximum integer item discrimination
maxiter	Maximum number of iterations

Value

Vector containing integer item discriminations

References

Verhelst, N. D. &, Glas, C. A. W. (1995). The one-parameter logistic model. In G. H. Fischer & I. W. Molenaar (Eds.). *Rasch Models* (pp. 215–238). New York: Springer.

Verhelst, N. D., Verstralen, H. H. F. M., & Eggen, T. H. J. M. (1991). *Finding starting values for the item parameters and suitable discrimination indices in the one-parameter logistic model*. CITO Measurement and Research Department Reports, 91-10.

See Also

See immer_cml for using immer_opcat to estimate the one-parameter logistic model.

Examples

```
#############################################################################
# EXAMPLE 1: Estimating integer item discriminations for dichotomous data
#############################################################################

library(sirt)
data(data.read, package="sirt")
dat <- data.read
I <- ncol(dat)

#--- estimate 2PL model
mod <- sirt::rasch.mml2( dat, est.a=1:I, mmliter=30)
summary(mod)
a <- mod$item$a        # extract (non-integer) item discriminations

#--- estimate integer item discriminations under different conditions
a1 <- immer::immer_opcat( a, hmean=3, min=1, max=6 )
table(a1)
a2 <- immer::immer_opcat( a, hmean=2, min=1, max=3 )
a3 <- immer::immer_opcat( a, hmean=1.5, min=1, max=2 )
#--- compare results
cbind( a, a1, a2, a3)
```

---

immer_proc_data *Processing Datasets and Creating Design Matrices for Rating Data*

Description

The function immer_proc_data processes datasets containing rating data into a dataset into a long format of pseudoitems (item $\times$ raters).

The function immer_create_design_matrix_formula creates a design matrix for a processed dataset and a provided formula.

Usage

```
immer_proc_data(dat, pid=NULL, rater=NULL, weights=NULL, maxK=NULL)

immer_create_design_matrix_formula( itemtable, formulaA )
```

Arguments

dat	Datasets with integer item responses
pid	Vector with person identifiers
rater	Vector with rater identifiers
weights	Vector with sampling weights
maxK	Optional vector with maximum category per item

itemtable	Processed item table. The table must include the column item (an integer item identifier) and maxK (maximum number of categories per item). Optional columns are rater (an integer rater identifier),item_name and rater_name.
formulaA	An R formula. The facets item, step and rater are treated as numeric. However, numeric transformation can be applied for the step parameter by using the arguments item_num, step_num or rater_num in formulaA.

Value

The output of immer_proc_data is a list with several entries (selection)

dat2	Dataset containing pseudoitems
dat2.resp	Dataset containing response indicators for pseudoitems
dat2.NA	Dataset containing pseudoitems and missing responses coded as NA
dat	Original dataset
person.index	Person identifiers
rater.index	Rater identifiers
VV	Number of items
N	Number of persons
RR	Number of raters
dat2.ind.resp	Array containing indicators of pseudoitems and categories
ND	Number of person-rater interactions
itemtable	Information about processed data

The output of immer_create_design_matrix_formula is a list with several entries (selection)

A	design matrix
itemtable2	Processed item table

Examples

```
#############################################################################
# EXAMPLE 1: Processing rating data
#############################################################################

data(data.immer01a, package="immer")
dat <- data.immer01a

res <- immer::immer_proc_data( dat=dat[,paste0("k",1:5)], pid=dat$idstud,
            rater=dat$rater)
str(res, max.level=1)

## Not run:
#############################################################################
# EXAMPLE 2: Creating several design matrices for rating data
#############################################################################
```

```
data(data.ratings1, package="sirt")
dat <- data.ratings1
resp <- dat[,-c(1,2)]
#- redefine the second and third item such that the maximum category score is 2
for (vv in c(2,3)){
    resp[ resp[,vv] >=2,vv ] <- 2
}

#--- process data
res0 <- immer::immer_proc_data( dat=resp, pid=dat$idstud,  rater=dat$rater)

#--- rating scale model
des1 <- immer::immer_create_design_matrix_formula( itemtable=res0$itemtable,
                formulaA=~ item + step )
des1$des

#--- partial scale model
des2 <- immer::immer_create_design_matrix_formula( itemtable=res0$itemtable,
                formulaA=~ item + item:step )
des2$des

#--- multi-facets Rasch model
des3 <- immer::immer_create_design_matrix_formula( itemtable=res0$itemtable,
                formulaA=~ item + item:step + rater )
des3$des

#--- polytomous model with quadratic step effects
des4 <- immer::immer_create_design_matrix_formula( itemtable=res0$itemtable,
                formulaA=~ item + item:I(step_num^2) )
des4$des

## End(Not run)
```

---

immer_reshape_wideformat

Creating a Rating Dataset in Wide Format

Description

Converts a rating dataset from a long format into a wide format.

Usage

```
immer_reshape_wideformat(y, pid, rater, Nmin_ratings=1)
```

Arguments

y	Vector or a data frame containing ratings
pid	Person identifier

rater	Rater identifier
Nmin_ratings	Minimum number of ratings used for selection

Value

Data frame with ratings. Each row corresponds to a person, and each of the columns (except the first one containing the person identifier) to one rater.

Examples

```
#############################################################################
# EXAMPLE 1: Reshaping ratings of one variable into wide format
#############################################################################

data(data.immer03)
dat <- data.immer03

# select variable "b" and persons which have at least 10 ratings
dfr <- immer::immer_reshape_wideformat( y=dat$b2, pid=dat$idstud, rater=dat$rater,
                    Nmin_ratings=10 )
head(dfr)

#############################################################################
# EXAMPLE 2: Reshaping ratings of a data frame into wide format
#############################################################################

data(data.immer07)
dat <- data.immer07

#*** Dataset 1: Wide format for item I1
dfr1 <- immer::immer_reshape_wideformat( dat$I1, rater=dat$rater, pid=dat$pid)

#*** Dataset 2: Wide format for four items I1, I2, I3 and I4
dfr2 <- immer::immer_reshape_wideformat( dat[, paste0("I",1:4) ],
              rater=dat$rater, pid=dat$pid )
str(dfr2)
```

---

immer_unique_patterns *Extracts Unique Item Response Patterns*

Description

Extracts unique item response patterns.

Usage

```
immer_unique_patterns(dat, w=rep(1, nrow(dat)))
```

Arguments

dat	Data frame containing integer item responses
w	Optional vector of weights

Value

A list with entries

y	Data frame with unique item response patterns
w	Vector of frequency weights
y_string	Item response pattern coded as a string

See Also

See `mirt::expand.table` for back-converting unique item response patterns into a data frame with item responses.

Examples

```
#############################################################################
# EXAMPLE 1: Unique item response patterns data.read
#############################################################################

data( data.read, package="sirt")
dat <- data.read

# extract item response patterns
res <- immer::immer_unique_patterns(dat)

## Not run:
# back-conversion with expand.table
dat2 <- mirt::expand.table( cbind( res$y, res$w ) )
# check correctness
colMeans(dat)
colMeans(dat2)

## End(Not run)
```

`lc2_agreement` *A Latent Class Model for Agreement of Two Raters*

Description

Estimates a latent class model for agreement of two raters (Schuster & Smith, 2006). See Details for the description of the model.

Usage

```
lc2_agreement(y, w=rep(1, nrow(y)), type="homo", method="BFGS", ...)

## S3 method for class 'lc2_agreement'
summary(object, digits=3,...)

## S3 method for class 'lc2_agreement'
logLik(object, ...)

## S3 method for class 'lc2_agreement'
anova(object, ...)
```

Arguments

`y`	A data frame containing the values of two raters in columns
`w`	Optional vector of weights
`type`	Type of model specification. Can be `"unif"`, `"equal"`, `"homo"` or `"hete"`. See Details.
`method`	Optimization method used in `stats::optim`
`...`	Further arguments passed to `stats::optim`
`object`	Object of class `l2_agreement`
`digits`	Number of digits for rounding

Details

The latent class model for two raters decomposes a portion of ratings which conform to true agreement and another portion of ratings which conform to a random rating of a category. Let X_r denote the rating of rater r, then for $i \neq j$, it is assumed that

$$P(X_1 = i, X_2 = j) = \phi_{1i}\phi_{2j}(1 - \gamma)$$

For $i = j$ it is assumed that

$$P(X_1 = i, X_2 = i) = \tau_i \gamma + \phi_{1i}\phi_{2i}(1 - \gamma)$$

where γ denotes the proportion of true ratings.

All τ_i and ϕ_{ri} parameters are estimated using `type="hete"`. If the ϕ parameters are assumed as invariant across the two raters (i.e. $\phi_{1i} = \phi_{2i} = \phi_i$), then `type="homo"` must be specified. The constraint $\tau_i = \phi_i$ is imposed by `type="equal"`. All ϕ_i parameters are set equal to each other using `type="unif"`.

Value

`model_output`	Output of the fitted model
`saturated_output`	Output of the saturated model
`LRT_output`	Output of the likelihood ratio test of model fit

partable	Parameter table
parmsummary	Parameter summary
agree_true	True agreement index shich is the γ parameter
agree_chance	Agreement by chance
rel_agree	Conditional reliability of agreement
optim_output	Output of optim from the fitted model
nobs	Number of observations
type	Model type
ic	Information criteria
loglike	Log-likelihood
npars	Number of parameters
y	Used dataset
w	Used weights

References

Schuster, C., & Smith, D. A. (2006). Estimating with a latent class model the reliability of nominal judgments upon which two raters agree. *Educational and Psychological Measurement, 66*(5), 739-747.

Examples

```
#############################################################################
# EXAMPLE 1: Dataset in Schuster and Smith (2006)
#############################################################################

data(data.immer08)
dat <- data.immer08

# select ratings and frequency weights
y <- dat[,1:2]
w <- dat[,3]

#*** Model 1: Uniform distribution phi parameters
mod1 <- immer::lc2_agreement( y=y, w=w, type="unif")
summary(mod1)

#*** Model 2: Equal phi and tau parameters
mod2 <- immer::lc2_agreement( y=y, w=w, type="equal")
summary(mod2)

## Not run:
#*** Model 3: Homogeneous rater model
mod3 <- immer::lc2_agreement( y=y, w=w, type="homo")
summary(mod3)

#*** Model 4: Heterogeneous rater model
```

```
mod4 <- immer::lc2_agreement( y=y, w=w, type="hete")
summary(mod4)

#--- some model comparisons
anova(mod3,mod4)
IRT.compareModels(mod1,mod2,mod3,mod4)

## End(Not run)
```

---

probs2logits *Conversion of Probabilities into Logits*

Description

Converts probabilities into logits

Usage

```
probs2logits(probs)

logits2probs(y)
```

Arguments

probs	Vector containing probabilities
y	Vector containing logits

Value

A vector with logits or probabilities

Examples

```
#############################################################################
# EXAMPLE 1: Probability-logit-conversions: a toy example
#############################################################################

# define vector of probabilities
probs <- c( .3, .25, .25, .2)
sum(probs)

# convert probabilities into logits
y <- immer::probs2logits( probs )
# retransform logits into probabilities
immer::logits2probs(y)
```

Index