

Package ‘mixRasch’

February 14, 2012

Type Package

Title Mixture Rasch Models with JMLE

Version 0.1

Date 2009-03-13

Author John T. Willse

Maintainer John T. Willse <willse@uncg.edu>

Description Estimates mixture Rasch models, including the dichotomous Rasch model, the rating scale model, and the partial credit model.

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-03-19 08:02:27

R topics documented:

mixRasch-package	2
mixRasch	2
SimMix	5
Index	6

mixRasch-package	<i>Contains a function for estimating and evaluating mixture Rasch models using JMLE</i>
------------------	--

Description

The included function will estimate a mixture Rasch model using joint maximum likelihood estimation (JMLE). The estimation is based on a mixture partial credit model. Step parameters can be constrained to estimate a mixture rating scale model. Estimating a model with only one latent class accomplishes a standard Rasch analysis with JMLE.

Details

Package:	mixRasch
Type:	Package
Version:	0.1
Date:	2009-03-13
License:	GPL version 2 or newer

This is an early version of the package. It currently contains only the mixRasch function. Please contact the author if you encounter any bugs or if you have questions or suggestions.

Author(s)

John T. Willse <jtwillse@uncg.edu>

References

Willse, J. T. (2009). Mixture Rasch models with joint maximum likelihood estimation. Presented at the National Council on Measurement in Education annual meeting. San Diego.

mixRasch	<i>Function for estimating and evaluating mixture Rasch models using JMLE</i>
----------	---

Description

This function will estimate a mixture Rasch model using joint maximum likelihood estimation (JMLE). The estimation is based on a mixture partial credit model. Step parameters can be constrained to estimate a mixture rating scale model.

Usage

```
mixRasch(data, steps, max.iter = 50, conv.crit = 0.001, model = "RSM",
         n.c = 1, class, metric, info.fit = TRUE, treat.extreme = 0.3,
         maxchange = 1.5, maxrange = c(-4, 4), as.LCA = FALSE)
```

Arguments

<code>data</code>	A rectangular data set (matrix or data frame) to be analyzed.
<code>steps</code>	The maximum number of item thresholds to be estimated. Some items may have less than the maximum.
<code>max.iter</code>	Maximum number EM iterations
<code>conv.crit</code>	Estimation stops when the largest model parameter change is smaller than this criterion.
<code>model</code>	"RSM" (the default) constrains all step parameters to be equal (i.e., estimates a rating scale model). Assumes all items have the same number of steps. "PCM" allows step parameters to differ across items (i.e., estimates a partial credit model). The number of steps can differ across items.
<code>n.c</code>	Number of latent classes.
<code>class</code>	Optional matrix of starting values for latent class membership.
<code>metric</code>	Not implemented. Will be an optional argument for setting the final scale of the Rasch results.
<code>info.fit</code>	If "True" the information based criteria of fit (AIC, BIC) are estimated.
<code>treat.extreme</code>	Adjustment to perfect response vectors to allow estimation of person parameters. Perfect vectors are not used during item parameter estimation.
<code>maxchange</code>	Limits the change to model parameters in a single iteration. Helps keep estimates reasonable in the first few iterations.
<code>maxrange</code>	Admissible range of item difficulties.
<code>as.LCA</code>	If TRUE, all person parameters are constrained to equal zero. That analysis accomplishes a latent class analysis rather than a mixture Rasch model.

Details

The analyzed data should be a rectangular data file. Missing data are permitted and handled in the usual JMLE manner. Large amounts of missing data will slow down estimation. Basic results on item parameters are returned to the screen. When more than one class is specified (`n.c` greater than 1) Rasch model parameters and measures of fit are returned in a list, with each class's results returned in a separate element of that list. When `n.c = 1` a standard Rasch model analysis is performed.

Value

<code>LatentClass</code>	One element for each latent class. Each <code>LatentClass</code> element is itself a list containing within class results: item statistics (<code>i.stat</code>), person parameters and measures of person fit (<code>person.par</code>), item parameters and measures of item fit (<code>item.par</code>).
--------------------------	---

max.change	The largest model parameter change in each class in the last iteration of the estimation.
class	A rectangular matrix of the order number of people by number of classes. Values represent the probability that a person (row) belongs to a particular class (column).
iter	The number of completed iterations.
converge.flag	Indicates if convergence was reached.
info.fit	Provides (if requested) AIC, BIC, CAIC, log likelihood, number of estimated parameters, and number of persons used in the estimation.
model	Reports which model was estimated.
removed.items	Provides a vector indicating which, if any, items had to be removed from estimation for having perfect vectors.
run.time	Reports time elapsed during estimation.

Note

Be aware that this function is an early implementation of the procedure. Please contact the author if you encounter any bugs or if you have questions or suggestions.

Author(s)

John T. Willse

References

Willse, J. T. (2009). Mixture Rasch models with joint maximum likelihood estimation. Presented at the National Council on Measurement in Education annual meeting. San Diego.

Examples

```
# Example data included with mixRasch
data(SimMix)

test1 <- mixRasch(SimMix,1,50, conv.crit=.0001, n.c=1)
test2 <- mixRasch(SimMix,1,500, conv.crit=.0001, n.c=2)

# The 3 class solution may take a while to converge or need
# to be restarted if it fails to converge.
test3 <- mixRasch(SimMix,1,1000, conv.crit=.0001, n.c=3)

# Notice that the AIC and BIC are lowest for the 2 class solution
rbind(test1$info.fit,test2$info.fit,test3$info.fit)

# Notice that the two "difficulty" columns are ordered differently
# The results reflect that the two groups in the data set have a reversed
# scale from one another.
test2
```

`SimMix`*Example 2-class data*

Description

This simulated data contains two groups. The item difficulties for the two groups are reversed. The analysis should make a clear example for those people less familiar with mixture models.

Usage

```
data(SimMix)
```

Format

A data frame with 1000 observations on the following 10 variables.

- i1 a numeric vector
- i2 a numeric vector
- i3 a numeric vector
- i4 a numeric vector
- i5 a numeric vector
- i6 a numeric vector
- i7 a numeric vector
- i8 a numeric vector
- i9 a numeric vector
- i10 a numeric vector

Source

Simulated to follow a 2-class mixture Rasch model.

See Also

`mixRasch`

Examples

```
data(SimMix)
```

Index

*Topic **datasets**

SimMix, [5](#)

*Topic **misc**

mixRasch, [2](#)

*Topic **package**

mixRasch-package, [2](#)

mixRasch, [2](#)

mixRasch-package, [2](#)

SimMix, [5](#)