

The MERpsychophysics Reference Manual

Version 1.0

Date 22.05.2012

Authors: Alessandro Moscatelli

Maintainer: Alessandro Moscatelli <alessandro.moscatelli@uni-bielefeld.de>

Requires R(>= 2.11), lme4, languageR, mnormt, Matrix, lattice

License: GPL (>=2)

Description: The software provides a supplement to the R package lme4 for modelling and plotting psychophysical data with Generalized Linear Mixed Models. For an example with simulated data run MERdemo.r. Further information will be soon available in the author's web page (<http://mixedpsychophysics.wordpress.com/>)

The MERpsychophysics (an R based software)

Description

The software provides a supplement to the R package lme4 for modelling and plotting psychophysical data with Generalized Linear Mixed Models.

Details

Version: 1.0

Date: 01.01.2012

License: GNU public license

The software provides functions to estimate the Point of Subjective Equality (PSE) and the Just Noticeable Difference (JND) within the GLMM framework. All functions require as input a GLMM fitted with `glmer{lme4}` (a "mer" object, see package lme4 for further details). Currently, the code only allows probit models (corresponding to the cumulative Gaussian distribution in the psychometric function). The functions provided by MERpsychophysics are the following:

- *MERplot* plots a mer object as psychometric functions (one for each cluster/subject).
- *MERboot* estimates the PSE, the JND and their confidence interval by means of bootstrap method. The other two functions *resample.mer* and *suff.GLMM* are called by MERboot for resampling and ordering the output.
- *delta.psy.probit* estimates the PSE, the JND and their confidence interval by means of delta method. This function can be used on glm object (lme4 = F) or on a mer object (lme4 = T) having a probit link function.
- *MERsimulate* simulates a dataset from a psychophysical experiment.

The following packages should be installed, as ancillary functions in this package depend on them:

- lme4
- mnormt
- Matrix
- Lattice
- MASS

The package *languageR* contains useful functions for inference a plotting of mer objects. We also suggest the package *psyphy*, which contains useful functions for the analysis of the single subject.

Author(s)

Alessandro Moscatelli

University of Bielefeld, Bielefeld, Germany

<alessandro.moscatelli@uni-bielefeld.de>

Maura Mezzetti

University of Rome "Tor Vergata", Rome, Italy

Francesco Lacquaniti

University of Rome "Tor Vergata", Rome, Italy

References

[...]

Examples

```
#setwd("/mypath1/MERpsychophysics")
source("MERpsychophysics.r")

#Simulate a dataset (default values)
datafr <- MERsimulate(nsubject = 5, constant = F)

#fit the GLMMwith glmer (probit link function)
mod1 <- glmer(cbind(Longer, Total - Longer) ~ X + (1 + X| Subject), family =
binomial(link = "probit"), data = datafr, nAGQ = 10)

#Estimate of PSE and JND via botstrap method and delta method
MERboot(mod1, size = 200, balance = T)
delta.psy.probit(mod1, lme4 = T)

#Plot the figure
MERplot(mod1, x.from = 250, x.to = 1350, col = T)
```

Function: delta.psy.probit Confidence interval of the PSE and the JND with the Delta method

Description

Estimate of the PSE, the JND and their confidence interval by means of the delta method.

Usage

```
delta.psy.probit(model, alpha = 0.05, lme4 = F)
```

Arguments

model	a glm or mer object. The default is a glm object, if using a mer object set lme4 = T (see below)
alpha	the confidence level around the estimate of the parameter. If alpha = 0.05 (default value), the true parameter is within the confidence interval with a probability of 0.95.
lme4	logical. Mixed-effects model fitted with lme4 (lme = T) or fixed-effects model (ordinary GLM)?

Details

The PSE and the JND are functions of the intercept and slope of a GLM/GLMM (fixed effects intercept and slope for GLMM). The function delta.psy.probit estimates the parameters and their confidence interval by means of the delta method. The function assumes that the intercept and the slope are respectively the first and the second coefficient of the GLM/GLMM. In case of model having multiple predictors, pay attention to the order of the coefficients in the formula.

Value

A matrix with estimate, standard error and confidence interval of the PSE and the JND.

References

Faraggi et al. (2003) Statistics in Medicine 22:1977-1988

Examples

See MERdemo.r

Function: MERboot PSE, JND and corresponding CI estimated with bootstrap method

Description

Estimate of the PSE, the JND and their confidence intervals by means of the bootstrap method.

Usage

```
MERboot(model, size = 600, confint = 0.95, fileout = F, filenamePSE = "MCMCpse.pdf", filenameJND = "MCMCjnd.pdf")
```

Arguments

model	a mer object with a single fixed-effect variable and one or two random- effects variable. The function assumes the GLMM has a probit link function (that is, a cumulative Gaussian model).
size	number of simulated dataset in the bootstrap procedure. Default value equal 600. The reliability of the estimate increases with the size value (although this increase the time required by the analysis)
confint	A number ranging from 0 to 1, giving the probability covered by the confidence interval. Default value is 0.95 (that is, Inferior and Superior CI cover the 95% probability of the bootstrap distribution)
fileout	logical. If True, two pdf files are generated with the histogram of bootstrap samples, the median and confidence interval.
filenamePSE, filenameJND	respectively, the file name of pdf files for the two parameters

Details

This bootstrap method takes into account both fixed and random effects of the model for the estimate. The algorithm first simulates the random predictors (the between-subjects random effects) form a Normal distribution. Using these random predictors and the fitted fixed effects, it simulates the responses from a binomial distribution, and it fit again the GLMM. This simulation is repeated a number of times equal to $B = \text{size}$, providing the bootstrap estimates.

In the current version of the function, the responses fitted with lme4 must be binned with respect to the stimulus level and the participant, for example:

Subject ID	Longer	Total	Stimulus
S1	0	20	10
S1	5	20	15
S1	10	20	20
S1	15	20	25
S1	20	20	30
S2	1	20	10
S2	6	20	15
S2	10	20	20
...

The number of trials in Total column can be different from one participant to the other, but the levels of the stimulus (and therefore, the number of points in the psychometric function) has to be the same in all participants. In the example above, all participants are presented with stimuli 10,15, 20, 25, and 30. The order of the columns is not relevant. Use MERsimulate to see an example of dataframe.

Value

A matrix with estimate, standard error and confidence interval of the PSE and the JND. Warnings may occur, due to false or singular convergence in `glmer{lme4}`. Usually, a singular or false convergence in few simulations doesn't affect the bootstrap estimate. We suggest replicating the analysis and checking that the estimate is stable. If necessary, increase the number of replications in the size parameter.

See Also

MERsimulate, resample.mer

References

[...]

Examples

See MERdemo.r

Function: MERplot Plot of GLMM and row data

Description

Plotting of a mer object (probit link) and row data as n psychometric functions (where n is the number of clusters or subjects)

Usage

```
MERplot(model, randcol = 3, x.label = "Stimulus Intensity", y.label = "Predicted Response", p05line = F, x.ref = mean(model@X[,2]), x.from = min(model@X[,2]), x.to = max(model@X[,2]), col = F)
```

Arguments

model	a mer object with a single fixed-effect variable and one or two random- effects variable. The function assumes the GLMM has a probit link function (that is, a cumulative Gaussian model).
randcol	The column of the random factor (i.e. the column with subject names). Check <code>mermodel@frame</code> to find the column in your mer model – randcol is equal to 3 if a single fixed-effect predictor has been fitted.
x.label, y.label	labels for x and y axes. Default labels are respectively “Stimulus Intensity” and “Predicted Response”.
p05line	logical. If it is TRUE, a dotted cross is plotted beyond the curves, pointing to 0.5 in the y-axis and to the value in x.ref in the x-axis
x.ref	the x value of the dotted cross. Default value is the mean of the continuous predictor.
x.from, x.to	the x coordinates range. Default values are the minimum and maximum of the continuous predictor. The y range is by default <code>c(0,1)</code> .
col	logical. If it is TRUE, a colour code is used to identify each subject.

Details

The model is by default a mer object with a single fixed-effect variable and one or two random- effects variable. In case of mer object with more than one fixed effect, only the first two fixed-effect coefficients (i.e. the intercept and the slope of the first fixed-effect predictor) will be taken into account for the plot. The function uses a cumulative Gaussian function for each subject, and therefore assumes that the beta coefficient of the fixed effect is > 0 (cumulative distributions are monotonically increasing). If the beta coefficients is < 0 , then the function is the complement of the cumulative distribution.

Value

The predictions of the intercept and slope for each subject or cluster (computed as the algebraic sum of the fixed effects and BLUPS). The plot is in the pdf file specified in filename.

See Also

`GLMMplot`

References

Baayen R. H., Davidson D. J., Bates D. M. (2008). Mixed-effects modelling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59, 390-412.

Examples

See `MERdemo.r`

Function: MERsimulate Simulation of a dataset

Description

The function simulates a dataset as if from a psychophysical experiment. Default values (parameters and stimulus range) are plausible in the psychophysics of time perception

Usage

MERsimulate (fixeff = c(-7, 0.00875), ranef = c(2.4, -20e-04, 2e-06), nsubjects = 8, pps = 9, ntrials = 40, xint = c(400, 1200), xby = 100, constant = F)

Arguments

fixeff	a vector of length 2 of fixed effects. Details on the model used for the generation of random samples are given under 'Details'
ranef	a vector of length 3 containing in order the Variance of the Random Intercept, the Covariance and the Variance of the Random Slope
nsubjects	an integer for the number of subjects in the dataset
pps	an integer for the number of points per subject in the psychometrics curve
ntrials	an integer for the number of trials at each point of the psychometric function
xint	a vector of length 2 giving the upper and lower interval of the independent variable
constant	logical. If True, stimuli are equally spaced, the distance is equal to xby. If False, the stimuli are sampled at random from a uniform distribution.
xby	if constant = T, xby is the distance between stimuli

Details

The function generates random data frame within the general structure of a GLMM (probit link). We first define the fixed Intercept and slope and the Variance and Covariance matrix of the random effects. We find the nearest positive definite matrix of this Variance and Covariance matrix and use it to randomly generate the adjustments for each subject (corresponding to BLUPS in GLMM). In this way we get n probit models (where n is the number of subjects). For each value of independent variable X we find the corresponding probability (the inverse function of the probit is the cumulative Gaussian distribution). Then we sample the responses from a binomial distribution having p = probability and size = ntrials.

We remind the reader that the correlation between two random variables is

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{SD_X \cdot SD_Y}$$

And its range is (-1, 1). Chose the variance and covariance appropriately. Clearly, an unrealistic choice of the parameters will generate unrealistic responses.

Value

The function return a data.frame object of 6 columns and n rows, with the predictor variable, Intercept and Slope of each subject, responses (named 'Longer' as if from a time perception task), sample size and label of each subject.

See Also

MERboot, resample.mer

References

[...]

Example:

```
datafr = MERsimulate(nsubjects = 10)
```

Function: GLMMplot Plot of GLMM and row data

Description

Plotting of a GLMM (probit link) and row data as n psychometric functions (where n is the number of clusters or subjects)

Usage

```
GLMMplot (dataframe, X.col, Yes.col, Total.col, Subject.col, estimates, lme4 = F,  
model, x.label = "Stimulus Intensity", y.label = "Predicted Response", p05line = F,  
x.ref = mean(dataframe[,X.col]), x.from = min(dataframe[,X.col]), x.to =  
max(dataframe[,X.col]), col = F)
```

Arguments

dataframe	a data.frame object, containing the continuous experimental variable X, number of "Yes" responses, number of Total responses, subject identifier (either character, number or factor).
X.col, Yes.col, Total.col, Subject.col	integer labelling respectively the column of X the column of Yes responses, the column of Total responses and the column of Subject labels
estimates	a n-per-2 matrix with the intercept and slope for each subject (the sum of random and fixed predictor). If lme4 = T, the estimates are extracted from the mer object provided by <i>model</i>
lme4	logical. If TRUE, the matrix estimates is extracted from the mer object provided by <i>model</i>
model	a mer object with a single fixed-effect variable and one or two random- effects variable. The function assumes the GLMM has a probit link function (that is, a cumulative Gaussian model).
x.label, y.label	labels for x and y axes. Default labels are respectively "Stimulus Intensity" and "Predicted Response".
p05line	logical. If it is TRUE, a dotted cross is plotted beyond the curves, pointing to 0.5 in the y-axis and to the value in x.ref in the x-axis
x.ref	the x value of the dotted cross. Default value is the mean of the continuous predictor.

x.from, x.to the x coordinates range. Default values are the minimum and maximum of the continuous predictor. The y range is by default `c(0,1)`.
col logical. If it is TRUE, a colour code is used to identify each subject.

Details

The function is similar to `MERplot`, the main difference is that in `GLMMplot` is not necessary providing a mer object. Values may be passed in dataframe and estimates. Alternatively, it is possible to provide a mer object and estimates of parameter (not row data!) are extracted from this model. The model is by default a mer object with a single fixed-effect variable and one or two random-effects variable. In case of mer object with more than one fixed effect, only the first two fixed-effect coefficients (i.e. the intercept and the slope of the first fixed-effect predictor) will be taken into account for the plot. The function uses a cumulative Gaussian function for each subject. It therefore assumes that the beta coefficient of the fixed effect is > 0 (cumulative distributions are monotonically increasing). This is the standard in psychophysics.

Value

The dataframe, with color label and point character for each subject

See Also

`MERplot`

References

Baayen R. H., Davidson D. J., Bates D. M. (2008). Mixed-effects modelling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59, 390-412.

Examples

See `MERdemo.r`

Function: `resample.mer` Sampling data from MER object for Bootstrap analysis

Description

The function is called in `MERboot`; it simulates a number of datasets equal to `sample.size` using the parameters of a GLMM fitted with `glmer{lme4}`

Usage

```
resample.mer =(model, sample.size = 200)
```

Arguments

model a mer object including a single fixed effect, one or two random effects
sample.size the number of simulated dataset. The default value is 200

Details

The information about the number of subjects and trials are stored in the mer object (actually, subject and trial number in simulation is the same as in the original dataset, and it has to in order to generate a reliable confidence interval). The function simulates in each data frame a number of subjects equal to the one stored in the original mer object. Between-subjects random effects are estimated from a univariate Normal distribution for GLMMs with a single random effect, and from a bivariate Normal distribution for GLMMs with a two random effects. The function `rmnorm` from package *rnormt* is used to simulate the bivariate Normal random variable (check if this package has been installed in R).

Value

A data frame with 8 variables: the predictor variable, the estimated value of intercept and slope for each subject and sample, the counts of YES responses and total of trials, the sampled data frame number, an empty column (in future, this will contain a dummy variable for a categorical predictor), the label identifying each subject.

See Also

MERboot

References

[...]

Example

```
#simulate a dataset
datafr <- MERsimulate(nsubject = 5, constant = F)

#fit the GLMM(probit link function)
mod1 <- glmer(cbind(Longer, Total - Longer) ~ X + (1 + X| Subject),
              family = binomial(link = "probit"), data = datafr, nAGQ = 10)
sim.values <- resample.mer(mod1)
```

Function: suff.GLMM Estimates the two parameters PSE and JND in a list of mer objects

Description

The function is called in MERboot; it estimates the parameters PSE and JND in a list of mer objects

Usage

```
suff.GLMM =(models)
```

Arguments

models a list of mer objects, each mer object including a single fixed effect, one or two random effects

Details

In each mer object, the PSE and the JND are estimated respectively as:

$$PSE = -\frac{\alpha}{\beta}; JND = \frac{0.674}{\beta}$$

Where α, β are respectively the intercept and the slope of the model and 0.674 is the quantile of a Standard Normal Distribution at p equal to 0.75. The formula of the JND apply only for a probit (mixed) model.

Value

A data frame with the estimate of the parameters and identifier (1 for PSE and 0 for JND)

See Also

MERboot

References

[...]

Example

See in MERboot