Package 'metaSDTreg'

October 14, 2020

Title Regression Models for Meta Signal Detection Theory

Version 0.2.1

Description Regression methods for the meta-SDT model. The package implements methods for cognitive experiments of metacognition as described in Kristensen, S. B., Sandberg, K., & Bibby, B. M. (2020). Regression methods for metacognitive sensitivity. Journal of Mathematical Psychology, 94. <doi:10.1016/j.jmp.2019.102297>.

Depends R (>= 3.6), ordinal (>= 2019.12.10),

Imports maxLik (>= 1.3-4), truncnorm (>= 1.0-7), Matrix (>= 1.2.17)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Simon Bang Kristensen [aut, cre],
Kristian Sandberg [aut],
Bo Martin Bibby [aut]

Maintainer Simon Bang Kristensen <sbmkristensen@health.sdu.dk>

Repository CRAN

Date/Publication 2020-10-14 08:00:02 UTC

R topics documented:

`coef.metaSDTreg` *Coefficients method for metaSDTreg*

Description

Coefficients method for metaSDTreg

Usage

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

Arguments

`object`	An object of class metaSDTreg.
`...`	For future methods.

Value

A named vector of parameter estimates.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)
coef(fit)
```

`lines.predict_roc` *Lines of predicted ROC curve*

Description

Plot lines method for objects of class 'predict_roc'.

Usage

```
## S3 method for class 'predict_roc'
lines(x, thr = seq(-10, 10, length = 1000), ...)
```

Arguments

`x`	A 'predict_roc' object to plot.
`thr`	The sequence of thresholds parametrising the ROC curve, if this is a function. Default to a length 1000 sequence from -10 to 10.
`...`	Addtional arguments passed to `lines`.

Value

Invisible.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)

## Plot observed type 1 ROC points
plot(predict_roc(metadata, type = '1'), xlim = 0:1, ylim = 0:1)

## Add Model-predicted ROC curve (estimated from subset of data)
lines(predict_roc(fit, type = '1'))
```

logLik.metaSDTreg *Log-likelihood of metaSDTreg*

Description

Extract the log-likelihood from a metaSDTreg object.

Usage

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

Arguments

object	Object of class 'metaSDTreg'.
...	For further methods.

Value

Numeric, the likelihood at the maximum found by the optimisation procedure.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)
logLik(fit)
```

metaSDTcontrol *Control for metaSDTreg*

Description

Control function for metaSDTreg. The function allows passing arguments to the maxLik and clm procedure used for optimisation and starting values, respectively.

Usage

```
metaSDTcontrol(
  meth = "BFGS",
  start.vals = NULL,
  hessianMeth = TRUE,
  returnMaxLik = FALSE,
  loglikUse = c("R", "C++"),
  scoreFun = FALSE,
  c.maxLik = list(),
  c.clm = list(maxIter = 1000, maxModIter = 20),
  ...
)
```

Arguments

meth	Method used by maxLik to perform maximisation, which must support constrained optimisation, see maxLik. Defaults to 'BFGS'.
start.vals	Starting values for the maximisation algorithm. Any NA entries will be set to zero. Defaults to NULL, in which case starting values will be obtained from the partial proportional odds model as implemented in the package **ordinal** using clm.
hessianMeth	Method to compute the Hessian of the optimisation. Passed to the argument finalHessian in maxLik.
returnMaxLik	Should the function return the fit object from maxLik rather that the metaSDTreg object? Defaults to FALSE.
loglikUse	For testing purposes. The C++ option is not presently included. Should the procedure use the log-likelihood implemented in R (default), or that in C++? Option will be dropped in future versions.
scoreFun	For testing purposes. Should the optimisation procedure use the analytic score function? If FALSE (and 'meth' relies on gradients) numerical methods are used. Presently, only numerical methods are implemented. Defaults to FALSE. Option default may change in future versions.
c.maxLik	A list that is passed to the 'MaxControl' object in maxLik. The list should contain control arguments to be passed to the maximisation method used by maxLik (the default being maxBFGS). In particular, 'iterlim' is the number of iterations (default 200) and a larger number of 'printlevel' (default 0) prints more information from the maximisation procedure.
c.clm	A list that is passed to clm.control containing controls to be used by clm. The clm is used to determine starting values. By default, maxIter is set to 1000 and maxModIter is set to 20.
...	For future methods.

Value

A list of class 'metaSDTcontrol' containing the control arguments.

Examples

```
names(metaSDTcontrol())
```

`metaSDTdata` *Construct metaSDTdata*

Description

Constructor function for a metaSDTdata object.

Usage

```
metaSDTdata(data, type1, type2, signal)
```

Arguments

`data`	Data frame to be converted. Data should be in long format with one row corresponding to a single response.
`type1`	A string naming the variable containing the type 1 response, which should be an ordered factor with two levels where the first level corresponds to 'noise',
`type2`	A string naming the variable containing the ordinal type 2 response, which should be an ordered factor.
`signal`	A string naming the variable containing the signal.

Details

If type1 or type 2 is not an ordered factor, the function returns a warning. The function constructs a data frame containing variables named c('type1','type2','A','signal') along with any other variable in 'data' that is not given as an argument to the function. Because of this a warning is issued when the names c('type1','type2','A','signal') are present in 'data'.

Value

A data object of class 'metaSDTdata'. This has attributes 'L', the number of levels in the ordinal type 2 rating, and 'K' which is two times L (the number of levels of the ordinal variable 'A').

Examples

```
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

summary(metadata)
summary.data.frame(metadata)
```

`metaSDTreg` *Meta-SDT regression*

Description

Fit the meta SDT regression model.

Usage

```
metaSDTreg(
  formula,
  data,
  subset,
  na.action = na.fail,
  control = metaSDTcontrol()
)
```

Arguments

`formula`	Formula specifying the regression model. Presently, the left-hand side should be the ordinal variable A, while the right-hand side must contain the signal variable from the metaSDTdata object. Note that the variable 'signal' has a special interpretation in the model, see 'Details' below.
`data`	Data frame to fit the model on. Should be declared as metaSDTdata using the `metaSDTdata` function.
`subset`	Optional argument specifying a subset of the data to be used. A logical statement, which is evaluated in 'data'. Only rows with TRUE are used in the analysis.
`na.action`	Method for handling missing values, see na.action. Default is na.fail which stops the function with an error in the presence of missing values among variables entering 'formula'.
`control`	Optimisation control, see `metaSDTcontrol`.

Details

The input data frame should be of class 'metaSDTdata' as constructed by the function `metaSDTdata`. This will contain a variable, 'signal' which presently must be included on the right-hand side of the formula. The left hand side must be the ordinal variable 'A' also contained in the metaSDTdata object. The 'signal' variable has a special interpretation as its coefficients are the signal sensitivities (type 1 d-prime and the two response specific type 2 sensitivities meta-d-prime). Presently all other variables, as determined by `all.vars(update(formula, . ~ . -signal ))`, enter proportionally in the model, i.e. they share cofficients across response scale levels. See Kristensen & al. (2020) for details. This can be written formally as follows (in the notation of Kristensen & al. (2020)):

$$V_{\mathcal{N}} = V_{\mathcal{S}} = V,$$

and $\beta_{\mathcal{N}}$, $\beta_{\mathcal{S}}$ and β agree on all but the first entrance which is the signal sensitivities $d_{mathcalN}$, $d_{mathcalS}$ and d, respectively. Note that 'formula' specifies the mean of the latent variables and not the threshold model. Accordingly, attemps to remove the intercept in 'formula' will be ignored with a warning. The function fails when the left-hand side of 'formula' is not the ordinal variable 'A' and when 'data' is not a 'metaSDTdata' object. Future versions may be less defensive and simply issue a warning in these cases. Note that constrained optimisation is used to maximise the likelihood under ordinality of the thresholds. The variance-covariance matrix is not adjusted following the constrained estimation, cf. documentation of `maxLik`. When interpreting results from `summary.metaSDTreg()` care should be taken regarding the z-testor and associated p-value for threshold parameters, as the distribution of the statistic depends on the null hypothesis which will only be reasonable under special circumstances as in other types of ordinal models.

Value

An object of class 'metaSDTreg'. This is a list object containing,

- logLik: The log likelihood after optimisation.
- coefficients: Estimated coefficients.
- vcov: Variance-covariance matrix of the maxLik object.
- call: The call issued to metaSDTreg.
- na.act: The NA action.

References

Kristensen, S. B., Sandberg, K., & Bibby, B. M. (2020). Regression methods for metacognitive sensitivity. Journal of Mathematical Psychology, 94. <doi:10.1016/j.jmp.2019.102297>.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit function to data of first 20 replications per subject
fit_sub <- metaSDTreg(A ~ signal,
                      data=metadata,
                      subset = m <= 20)
summary(fit_sub)


## Fit model to estimate thresholds and sensitivities
fit <- metaSDTreg(A ~ signal,
            data=metadata)

## True values are
## c(tau.n.2=-0.5, tau.n.1=-0.2, theta=0,
##   tau.s.1=0.8, tau.s.2=1, d.n=0.5, d=1, d.s=0.75)
coef(fit)
```

plot.predict_roc *Plot predicted ROC curve*

Description

Plot method for objects of class 'predict_roc'.

Usage

```
## S3 method for class 'predict_roc'
plot(x, thr = seq(-10, 10, length = 1000), rline = TRUE, ...)
```

Arguments

x	A 'predict_roc' object to plot.
thr	The sequence of thresholds parametrising the ROC curve, if this is a function. Default to a length 1000 sequence from -10 to 10.
rline	Logical, should the line of random discrimination be added to the plot? Defaults to TRUE.
...	Addtional arguments passed to plot.

Details

If neither 'xlab' nor 'ylab' is passed to plot the function supplies default x- and y-axis labels based on the type of ROC curve.

Value

Invisible.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)

## Model-predicted signal-specific ROC curve
signalROC <- predict_roc(fit, type = 's')
plot(signalROC, type = 'l')
```

`points.predict_roc` *Points from predicted ROC curve*

Description

Plot points method for objects of class 'predict_roc'.

Usage

```
## S3 method for class 'predict_roc'
points(x, thr = seq(-10, 10, length = 1000), ...)
```

Arguments

`x`	A 'predict_roc' object to plot.
`thr`	The sequence of thresholds parametrising the ROC curve, if this is a function. Default to a length 1000 sequence from -10 to 10.
`...`	Addtional arguments passed to `lines`.

Value

Invisible.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)

## Plot observed type 1 ROC points
plot(predict_roc(metadata, type = '1'), xlim = 0:1, ylim = 0:1, pch = 'x')

## Add Model-predicted ROC curve (estimated from subset of data)
points(predict_roc(fit, type = '1'))
```

predict_roc *Generic predict_roc method*

Description

Predict ROC curve from signal detection theory model.

Usage

```
predict_roc(object, ...)
```

Arguments

object	Signal detection theory model.
...	Further arguments passed to methods.

Value

An object of class 'predict_roc'.

See Also

predict_roc.metaSDTreg, predict_roc.metaSDTdata.

predict_roc.metaSDTdata *Observed ROC points*

Description

The observed points of the ROC curve from a 'metaSDTdata' object.

Usage

```
## S3 method for class 'metaSDTdata'
predict_roc(object, type = c("1", "n", "s"), s0 = 0, s1 = 1, ...)
```

Arguments

object	A 'metaSDTdata' object from which to calculate observed ROC points.
type	The type of ROC curve to predict. A character string, where '1' requests the type 1 ROC curve, 'n' requests the type 2 noise-specific and 's' the type 2 signal-specific ROC curve.
s0	Numeric, the value of object$signal to regard as 'noise'. Defaults to 0.
s1	Numeric, the value of object$signal to regard as 'signal'. Defaults to 1.
...	For future methods

Details

Note that the type 1 ROC points arise by using each criterion in turn to decide between 'signal' and 'noise'. Since this involves also the type 2 thresholds, such a curve is also sometimes referred to as a 'pseudo' ROC curve.

Value

A matrix two-column matrix of class 'predict_roc' with one row of c(FA, HR) per threshold (FA: False Alarm rate, HR: Hit Rate).

References

Maniscalco, B., & Lau, H. (2014). Signal Detection Theory Analysis of Type 1 and Type 2 Data: Meta-d , Response-Specific Meta-d , and the Unequal Variance SDT Model. In S. M. Fleming, & C. D. Frith (Eds.), The Cognitive Neuroscience of Metacognition (pp. 25 66). : Springer Berlin Heidelberg.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Observed signal-specific ROC curve
signalROC <- predict_roc(metadata, type = 's')
```

---

predict_roc.metaSDTreg

Predicted ROC curve

Description

Predict ROC curves from metaSDTreg object.

Usage

```
## S3 method for class 'metaSDTreg'
predict_roc(object, type = c("1", "n", "s"), s0 = 0, s1 = 1, ...)
```

Arguments

`object`	An object of class `metaSDTreg`.
`type`	The type of ROC curve to predict. A character string, where '1' requests the type 1 ROC curve (the default), 'n' requests the type 2 noise-specific and 's' the type 2 signal-specific ROC curve.
`s0`	Numeric, the value of 'signal' to regard as 'noise'. Defaults to 0.
`s1`	Numeric, the value of 'signal' to regard as 'signal'. Defaults to 1.
`...`	For future methods

Details

The 'metaSDTreg' object given to the function must have named coefficients with names as they would be if `metaSDTreg` is run without user-supplied starting values.

A ROC curve is a 2-D curve parametrised by some x given by c(FA(x), HR(x)) where FA is the false alarm rate and HR is the hit rate. For example, for type 1 ROC,

$$FA(x) = 1 - pnorm(x - s0 * d),$$

$$HR(x) = 1 - pnorm(x - s1 * d),$$

where d is the signal sensitivity.

Note that the predicted ROC curve is for a reference individual in the regression, i.e. additional covariates are not entered into the ROC so that reparametrisation of the 'metaSDTreg' model is needed to change predictions.

Value

A function of class 'predict_roc' containing the appropriate ROC curve. This is a function of x which returns c(FA,HR), where FA is the false alarm rate and HR is the hit rate.

References

Maniscalco, B., & Lau, H. (2014). Signal Detection Theory Analysis of Type 1 and Type 2 Data: Meta-d , Response-Specific Meta-d , and the Unequal Variance SDT Model. In S. M. Fleming, & C. D. Frith (Eds.), The Cognitive Neuroscience of Metacognition (pp. 25 66). : Springer Berlin Heidelberg.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)

## Model-predicted signal-specific ROC curve
signalROC <- predict_roc(fit, type = 's')
```

`print.metaSDTdata` *Print method for metaSDT data*

Description

Print method for metaSDT data

Usage

```
## S3 method for class 'metaSDTdata'
print(x, ...)
```

Arguments

x	The metaSDTdata object.
...	Additional arguments passed to print.data.frame

Value

Invisible.

Examples

```
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')
print(metadata)
```

---

print.metaSDTreg *Print method for metaSDTreg*

Description

Print method for metaSDTreg

Usage

```
## S3 method for class 'metaSDTreg'
print(x, ...)
```

Arguments

x	An object of class metaSDTreg.
...	For future methods.

Value

Invisible.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
(fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20))
```

print.summary.metaSDTdata

Print method for a summary.metaSDTdata object

Description

Print method for a summary.metaSDTdata object

Usage

```
## S3 method for class 'summary.metaSDTdata'
print(x, ...)
```

Arguments

x	A metaSDTdata object.
...	For future methods.

Value

Invisible.

Examples

```
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')
summary(metadata)
```

print.summary.metaSDTreg

Print summary method for metaSDTreg

Description

Print summary method for metaSDTreg

Usage

```
## S3 method for class 'summary.metaSDTreg'
print(x, ...)
```

Arguments

x	An object of class summary.metaSDTreg.
...	For future methods.

Value

Invisible.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)
summary(fit)
```

---

simMetaData *Simulated metacognition experiment*

Description

Data is simulated from the meta-SDT model, formula (3)-(4) in Kristensen, Sandberg and Bibby (2020). The true values of the simulation parameters are c(tau.n.2=-0.5, tau.n.1=-0.2, theta=0, tau.s.1=0.8, tau.s.2=1, d.n=0.5, d=1, d.s=0.75).

Usage

```
simMetaData
```

Format

A data frame with 25000 obs. of 5 variables:

id Subject id.

m Observation id, within subject.

S Signal indicator (0 or 1).

resp Type 1 response (0 or 1).

conf Type 2 response, ordered (1, 2 or 3).

References

Kristensen, S. B., Sandberg, K., & Bibby, B. M. (2020). Regression methods for metacognitive sensitivity. Journal of Mathematical Psychology, 94. <doi:10.1016/j.jmp.2019.102297>.

summary.metaSDTdata *Summarise a metaSDTdata object as a cognitive experiment.*

Description

Summarise a metaSDTdata object as a cognitive experiment.

Usage

```
## S3 method for class 'metaSDTdata'
summary(object, ...)
```

Arguments

object	The metaSDTdata object.
...	Additional arguments. Presently not used.

Value

A list with class 'summary.metaSDTdata' containing summaries.

Examples

```
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')
summary(metadata)
```

summary.metaSDTreg *Summary method for metaSDTreg*

Description

Summary method for metaSDTreg

Usage

```
## S3 method for class 'metaSDTreg'
summary(object, ...)
```

Arguments

object	An object of class metaSDTreg.
...	For future methods.

Value

A list with class 'summary.metaSDTreg' containing summaries.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)
summary(fit)
```

---

vcov.metaSDTreg *Variance-covariance method for metaSDTreg*

Description

Variance-covariance method for metaSDTreg

Usage

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

Arguments

object	An object of class metaSDTreg.
...	For future methods.

Value

A matrix of variances and covariances.

Examples

```
## Declare simulated data as metaSDTdata
metadata <- metaSDTdata(simMetaData, type1='resp', type2='conf', signal='S')

## Fit model to subset of data
fit <- metaSDTreg(A ~ signal,
            data=metadata,
            subset = m <= 20)

## Standard errors
sqrt(diag(vcov(fit)))
```

Index