Package 'MAINT.Data'

December 31, 2016

Type Package

Title Model and Analyse Interval Data

Version 1.0.1

Date 2016-12-29

Author Pedro Duarte Silva <psilva@porto.ucp.pt>, Paula Brito <mpbrito.fep.up.pt>

Maintainer Pedro Duarte Silva <psilva@porto.ucp.pt>

Description Implements methodologies for modelling interval data by Normal and Skew-Normal distributions, considering appropriate parameterizations of the variance-covariance matrix that takes into account the intrinsic nature of interval data, and lead to four different possible configuration structures. The Skew-Normal parameters can be estimated by maximum likelihood, while Normal parameters may be estimated by maximum likelihood or robust trimmed maximum likelihood methods.

License GPL-2

LazyLoad yes

LazyData yes

Depends R (>= 3.1.0), Rcpp (>= 0.11.0), sn (>= 1.3.0)

Imports MASS, methods, miscTools, robustbase, rrcov, pcaPP

LinkingTo Rcpp, RcppEigen

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-12-31 16:47:30

R topics documented:

MAINT.Data-package *Modelling and Analizing Interval Data*

Description

MAINT.DATA implements methodologies for modelling Interval Data by Normal and Skew-Normal distributions, considering four different possible configurations structures for the variance-covariance matrix. It introduces a data class for representing interval data and includes functions and methods for parametric modelling and analysing of interval data. It performs maximum likelihood and trimmed maximum likelihood estimation, statistical tests, as well as (M)ANOVA and Discriminant Analysis.

Details

In the classical model of multivariate data analysis, data is represented in a data-array where n "individuals" (usually in rows) take exactly one value for each variable (usually in columns). Symbolic Data Analysis (see, e.g., Noirhomme-Fraiture and Brito (2011)) provides a framework where new variable types allow to take directly into account variability and/or uncertainty associated to each single "individual", by allowing multiple, possibly weighted, values for each variable. New variable types - interval, categorical multi-valued and modal variables - have been introduced.
We focus on the analysis of interval data, i.e., where elements are described by variables whose values are intervals. Parametric inference methodologies based on probabilistic models for interval variables are developed in Brito and Duarte Silva (2011) where each interval is represented by its midpoint and log-range,for which Normal and Skew-Normal (Azzalini and Dalla Valle (1996)) distributions are assumed. The intrinsic nature of the interval variables leads to special structures of the variance-covariance matrix, which are represented by four different possible configurations.
MAINT.DATA implements the proposed methodologies in R, introducing a data class for representing interval data; it includes functions for modelling and analysing interval data, in particular maximum likelihood and trimmed maximum likelihood (see, e.g. Hadi and Luceno (1997)) estimation, and statistical tests for the different considered configurations. Methods for (M)ANOVA and Discriminant Analysis (Duarte Silva and Brito (2015)) of this data class are also provided.

Package:	MAINT.Data
Type:	Package
Version:	1.0.1
Date:	2016-12-29
License:	GPL-2
LazyLoad:	yes
LazyData:	yes

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

Maintainer: Pedro Duarte Silva <psilva@porto.ucp.pt>

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Noirhomme-Fraiture, M., Brito, P. (2011), Far Beyond the Classical Data Models: Symbolic Data Analysis. *Statistical Analysis and Data Mining* **4**(2), 157–170.

Examples

```
# Create an Interval-Data object containing the intervals for 899 observations
# on the temperatures by quarter in 60 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8],VarNames=c("T1","T2","T3","T4"))

#Display the first and last observations

head(ChinaT)
tail(ChinaT)

#Print summary statistics

summary(ChinaT)

#Create a new data set considering only the Winter (1st and 4th) quarter intervals

ChinaWT <- ChinaT[,c(1,4)]

# Estimate normal distributuion parameters by maximum likelihood, assuming
# the classical (unrestricted) covariance configuration Case 1

ChinaWTE.C1 <- mle(ChinaWT,CovCase=1)
cat("Winter temperatures of China -- normal maximum likelhiood estimation results:\n")
print(ChinaWTE.C1)
cat("Standard Errors of Estimators:\n") ; print(stdEr(ChinaWTE.C1))

# Estimate normal distributuion parameters by maximum likelihood,
# assuming that one of the C2, C3 or C4 restricted covariance configuration cases hold

ChinaWTE.C234 <- mle(ChinaWT,CovCase=2:4)
cat("Winter temperatures of China -- normal maximum likelhiood estimation results:\n")
print(ChinaWTE.C234)
```

```
cat("Standard Errors of Estimators:\n") ; print(stdEr(ChinaWTE.C234))

# Estimate normal distributuion  parameters robustly by fast maximun trimmed likelihood,
# assuming that one of the C2, C3 or C4 restricted covariance configuration cases hold

## Not run:
ChinaWTE.C234 <- fasttle(ChinaWT,CovCase=2:4)
cat("Winter temperatures of China -- normal maximum trimmed likelhiood estimation results:\n")
print(ChinaWTE.C234)

# Estimate skew-normal distributuion  parameters

ChinaWTE.SkN <- mle(ChinaWT,Model="SKNormal")
cat("Winter temperatures of China -- Skew-Normal maximum likelhiood estimation results:\n")
print(ChinaWTE.SkN)
cat("Standard Errors of Estimators:\n") ; print(stdEr(ChinaWTE.SkN))

## End(Not run)

#MANOVA tests assuming that configuration case 1 (unrestricted covariance)
# or 3 (MidPoints independent of Log-Ranges) holds.

ManvChinaWT.C13 <- MANOVA(ChinaWT,ChinaTemp$GeoReg,CovCase=c(1,3))
cat("Winter temperatures of China -- MANOVA by geografical regions results:\n")
print(ManvChinaWT.C13)

#Linear Discriminant Analysis

ChinaWT.lda <- lda(ManvChinaWT.C13)
cat("Winter temperatures of China -- linear discriminant analysis results:\n")
print(ChinaWT.lda)
cat("lda Prediction results:\n")
print(predict(ChinaWT.lda,ChinaWT)$class)

## Not run:
#Estimate error rates by ten-fold cross-validation

CVlda <- DACrossVal(ChinaWT,ChinaTemp$GeoReg,TrainAlg=lda,
CovCase=BestModel(H1res(ManvChinaWT.C13)),CVrep=1)
summary(CVlda[,,"Clerr"])
glberrors <-
  apply(CVlda[,,"Nk"]*CVlda[,,"Clerr"],1,sum)/apply(CVlda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")

#Robust Quadratic Discriminant Analysis

ChinaWT.rqda <- Robqda(ChinaWT,ChinaTemp$GeoReg)
cat("Winter temperatures of China -- robust quadratic discriminant analysis results:\n")
print(ChinaWT.rqda)
cat("robust qda prediction results:\n")
print(predict(ChinaWT.rqda,ChinaWT)$class)

## End(Not run)
```

```
# Create an Interval-Data object containing the intervals for characteristics
# of 27 cars models.

Cars <- IData(CarsData[1:8],Seq="MidPLogR_VarbyVar",
  VarNames=c("Price","EngineCapacity","TopSpeed","Acceleration"))

#Display the first and last observations

head(Cars)
tail(Cars)

# Estimate normal distributuion parameters

CarsNE <- mle(Cars)
cat("Cars data -- normal maximum likelhiood estimation results:\n")
print(CarsNE)
cat("Standard Errors of Estimators:\n") ; print(stdEr(CarsNE))

# Estimate normal distributuion  parameters robustly by full maximum trimmed likelihood,

## Not run:
CarsTE <- fulltle(Cars)
cat("Cars data -- normal maximum trimmed likelhiood estimation results:\n")
print(CarsTE)

# Estimate parameters searching through normal and Skew-Normal distributions.

CarsNSNE <- mle(Cars,Model="NrmandSKN")
cat("Cars data  -- Maximum likelhiood estimation results:\n")
print(CarsNSNE)
cat("Standard Errors of Estimators:\n") ; print(stdEr(CarsNSNE))

## End(Not run)
```

`BestModel-methods` *Methods for function BestModel in Package 'MAINT.Data'*

Description

Selects the best model according to the chosen selection criterion (currently, BIC or AIC)

Usage

```
BestModel(ModE,SelCrit=c("IdtCrt","BIC","AIC"))
```

Arguments

ModE	An object of class IdtE representing the estimates of a model fitted to a data set of interval-value variables
SelCrit	The model selection criterion. "dtCrt" stands for the criterion originally, while "BIC" and "AIC" represent respectivelly the Bayesian and Akaike information criteria.

Value

An integer with the index of the model chosen by the selection criterion

---

CarsData *Cars Data Set*

Description

This data set consist of MidPoints and LogRanges of the intervals for four characteristics (Price, EngineCapacity, TopSpeed and Acceleration) of 27 cars models partitioned into four diffetent classes (Utilitarian,Berlina,Sportive and Luxury).

Usage

```
data(CarsData)
```

Format

A data frame containing 27 observations on 9 variables, the first eight with the MidPoint and LogRange of the interval characteristics for 27 car models, the last one a factor indicating the model class.

ChinaTemp *China Temperatures Data Set*

Description

This data set consist of the intervals of observed temperatures (Celsius scale) in each of the four quarters, Q_1 to Q_4, of the years 1974 to 1988 in 60 chinese metereological stations; one outlier observation (YinChuan_1982) has been discarded. The 60 stations belong to different regions in China, which therefore define a partition of the 899 stations-year combinations.

Usage

```
data(ChinaTemp)
```

Format

A data frame containing 899 observations on 9 variables, the first eight with the lower and upper bounds of the temperatures by quarter in the 899 stations-year combinations, the last one a factor indicating the geografical region of each station.

`coef--methods` *Methods for function coef in Package 'MAINT.Data'*

Description

S4 methods for function coef. As in the generic coef S3 'stats' method, these methods extract parameter estimates for the models fitted to Interval Data.

Usage

```
## S4 method for signature 'IdtNDE'
coef(object, selmodel=BestModel(object), ...)
## S4 method for signature 'IdtSNDE'
coef(object, selmodel=BestModel(object), ParType=c("Centr", "Direct", "All"), ...)
## S4 method for signature 'IdtNandSNDE'
coef(object, selmodel=BestModel(object),  ParType=c("Centr", "Direct", "All"), ...)
```

Arguments

`object`	An object representing a model fitted to interval data.
`selmodel`	Selected model from a list of candidate models saved in object.
`ParType`	Parameterization of the Skew-Normal distribution. Alternatives are "Centr" for centred parameters, "Direct" for direct parameters and "All", for both types of parameters. See Arellano-Valle and Azzalini (2008) for details.
`...`	Additional arguments for method functions.

Value

A vector of parameter estimates.

References

Arellano-Valle, R. B. and Azzalini, A. (2008): "The centred parametrization for the multivariate skew-normal distribution". Journal of Multivariate Analysis, Volume 99, Issue 7, 1362-1382.

See Also

`stdEr`, `vcov`

`ConfTests-class` *Class "Configuration Tests"*

Description

ConfTests contains a list of the results of statistical likelihood-ratio tests that evaluate the goodness-of-fit of restricted models against more general ones. Currently, the models implemented are those based on the Normal and Skew-Normal distributions, with the four alternative variance-covariance matrix configurations.

Slots

`TestRes`**:** List of test results; each element is an object of type LRTest, with the following components:

QuiSq: Value of the Qui-Square statistics corresponding to the performed test.
df: Degrees of freedom of the Qui-Square statistics.
pvalue: p-value of the Qui-Square statistics value, obtained from the Qui-Square distribution with df degrees of freedom.
H0logLik: Logarithm of the Likelihood function under the null hypothesis.
H1logLik: Logarithm of the Likelihood function under the alternative hypothesis.

`RestModels`**:** The restrited model (corresponding to the null hypothesis)

`FullModels`**:** The full model (corresponding to the alternative hypothesis)

Methods

show `signature(object = "ConfTests")`: show S4 method for the ConfTests-class

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

See Also

`mle`, `IData`, `LRTest`

da-methods *Discriminant Analysis of Interval Data*

Description

lda and qda perform linear and quadratic discriminant analysis of Interval Data based on classic estimates of a mixture of Gaussian models. Roblda and Robqda do the same using robust estimates of location and scatter. snda performs discriminant analysis of Interval Data based on estimates of mixtures of Skew-Normal models

Usage

```
## S4 method for signature 'IData'
lda( x, grouping, prior="proportions", CVtol=1.0e-5, egvtol=1.0e-10,
  subset=1:nrow(x), CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE, ... )

## S4 method for signature 'IdtMxtNDE'
lda(x, prior="proportions", selmodel=BestModel(x), egvtol=1.0e-10,
  silent=FALSE, ... )

## S4 method for signature 'IdtClMANOVA'
lda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IdtClMANOVA'
lda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IdtLocNSNMANOVA'
lda( x, prior="proportions",
  selmodel=BestModel(H1res(x)@NMod), egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IData'
qda( x, grouping, prior="proportions", CVtol=1.0e-5, subset=1:nrow(x),
  CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE, ... )

## S4 method for signature 'IdtMxtNDE'
qda(x, prior="proportions", selmodel=BestModel(x), silent=FALSE,
  ... )

## S4 method for signature 'IdtHetNMANOVA'
qda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  silent=FALSE, ... )

## S4 method for signature 'IdtGenNSNMANOVA'
qda( x, prior="proportions",
  selmodel=BestModel(H1res(x)@NMod), silent=FALSE, ... )
```

```
## S4 method for signature 'IData'
Roblda( x, grouping, prior="proportions", CVtol=1.0e-5, egvtol=1.0e-10,
  subset=1:nrow(x), CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE,
  CovEstMet=c("Pooled","Globdev"), SngDMet=c("fasttle","fulltle"),
  Robcontrol=RobEstControl(), ... )

## S4 method for signature 'IData'
Robqda( x, grouping, prior="proportions", CVtol=1.0e-5,
  subset=1:nrow(x), CovCase=1:4, SelCrit=c("BIC","AIC"), silent=FALSE,
  SngDMet=c("fasttle","fulltle"), Robcontrol=RobEstControl(), ... )

## S4 method for signature 'IData'
snda(x, grouping, prior="proportions", CVtol=1.0e-5, subset=1:nrow(x),
  CovCase=1:4, SelCrit=c("BIC","AIC"), Mxt=c("Loc","Gen"), ... )

## S4 method for signature 'IdtLocSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IdtLocNSNMANOVA'
snda( x, prior="proportions",
  selmodel=BestModel(H1res(x)@SNMod), egvtol=1.0e-10, silent=FALSE, ... )

## S4 method for signature 'IdtGenSNMANOVA'
snda( x, prior="proportions", selmodel=BestModel(H1res(x)),
  silent=FALSE, ... )

## S4 method for signature 'IdtGenNSNMANOVA'
snda( x, prior="proportions",
  selmodel=BestModel(H1res(x)@SNMod), silent=FALSE, ... )
```

Arguments

x	An object of class IData, IdtLocSNMANOVA, IdtLocNSNMANOVA,IdtGenSNMANOVA or IdtGenNSNMANOVA with either the original Interval Data, or the results of a Interval Data Skew-Normal MANOVA, from which the discriminant analysis will be based.
grouping	Factor specifying the class for each observation.
prior	The prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
CVtol	Tolerance level for absolute value of the coefficient of variation of non-constant variables. When a MidPoint or LogRange has an absolute value within-groups coefficient of variation below CVtol, it is considered to be a constant.
subset	An index vector specifying the cases to be used in the analysis.

CovCase	Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit	The model selection criterion.
silent	A boolean flag indicating wether a warning message should be printed if the method fails.
CovEstMet	Method used to estimate the common covariance matrix in Roblda (Robust linear discriminant analysis). Alternatives are "Pooled" (default) for a pooled average of the the robust within-groups covariance estimates, and "Globdev" for a global estimate based on all deviations from the groups multivariate l1 medians. See Todorov and Filzmoser (2009) and pcaPP.l1median for details.
SngDMet	Algorithm used to find the robust estimates of location and scatter. Alternatives are "fasttle" (default) and "fulltle".
Robcontrol	A control object (S4) of class RobEstControl-class containing estimation options - same as these provided in the function specification. If the control object is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object.
Mxt	Indicates the type of mixing distributions to be considered. Current alternatives are "Hom" (homocedastic) and "Het" (hetereocedasic) for Gaussian models, "Loc" (location model – groups differ only on their location parameters) and "Gen" "Loc" (general model – groups differ on all parameters) for Skew-Normal models.
selmodel	Selected model from a list of candidate models saved in object x.
egvtol	Tolerance level for the eigenvalues of the product of the inverse within by the between covariance matrices. When a eigenvalue has an absolute value below egvtol, it is considered to be zero.
...	Other named arguments.

References

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

IData, IdtLocSNMANOVA,IdtLocSNMANOVA,IdtLocSNMANOVA,IdtLocSNMANOVA, pcaPP.l1median.

Examples

```
# Create an Interval-Data object containing the intervals for 899 observations
# on the temperatures by quarter in 60 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8],VarNames=c("T1","T2","T3","T4"))
```

```
#Linear Discriminant Analysis

ChinaT.lda <- lda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- linear discriminant analysis results:\n")
print(ChinaT.lda)
cat("lda Prediction results:\n")
print(predict(ChinaT.lda,ChinaT)$class)

## Not run:
#Estimate error rates by ten-fold cross-validation replicated 20 times

CVlda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=lda,CovCase=CovCase(ChinaT.lda))
summary(CVlda[,,"Clerr"])
glberrors <-
  apply(CVlda[,,"Nk"]*CVlda[,,"Clerr"],1,sum)/apply(CVlda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")

## End(Not run)

#Quadratic Discriminant Analysis

ChinaT.qda <- qda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- qda discriminant analysis results:\n")
print(ChinaT.qda)

## Not run:
#Estimate error rates by ten-fold cross-validation replicated 20 times

CVqda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=qda,CovCase=CovCase(ChinaT.qda))
summary(CVqda[,,"Clerr"])
glberrors <-
  apply(CVqda[,,"Nk"]*CVqda[,,"Clerr"],1,sum)/apply(CVqda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")

# Skew-Normal based discriminant analysis, asssuming that the different regions may differ
# in all SkewNormal parameters

cat("Temperatures of China -- SkewNormal general model discriminant analysis results:\n")
ChinaT.gensnda <- snda(ChinaT,ChinaTemp$GeoReg,Mxt="Gen")
print(ChinaT.gensnda)

#Estimate error rates by three-fold cross-validation without replication

CVgensnda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=snda,Mxt="Gen",
  CovCase=CovCase(ChinaT.gensnda),kfold=3,CVrep=1)
summary(CVgensnda[,,"Clerr"])
glberrors <-
  apply(CVgensnda[,,"Nk"]*CVgensnda[,,"Clerr"],1,sum)/apply(CVgensnda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")

#Robust Quadratic Discriminant Analysis
```

```
ChinaT.rqda <- Robqda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- robust qda discriminant analysis results:\n")
print(ChinaT.rqda)

#Estimate error rates by ten-fold cross-validation with 5 replications

CVrqda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=Robqda,CovCase=CovCase(ChinaT.rqda),
   CVrep=5)
summary(CVrqda[,,"Clerr"])
glberrors <-
  apply(CVrqda[,,"Nk"]*CVrqda[,,"Clerr"],1,sum)/apply(CVrqda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")


## End(Not run)
```

DACrossVal *Cross Validation for Discriminant Analysis Classification Rules*

Description

'DACrossVal' evaluates the performance of a Discriminant Analysis training sample algorithm by k-fold Cross-Validation.

Usage

```
DACrossVal(data, grouping, TrainAlg, EvalAlg=EvalClrule,
Strfolds=TRUE, kfold=10, CVrep=20, prior="proportions", loo=FALSE, ...)
```

Arguments

data	Matrix, data frame or Interval Data object of observations.
grouping	Factor specifying the class for each observation.
TrainAlg	A function with the training algorithm. It should return an object that can be used as input to the argument of 'EValAlg'.
EvalAlg	A function with the evaluation algorithm. By default set to 'EvalClrule' which returns a list with components "err" (estimates of error rates by class) and "Nk" (number of out-sample observations by class). This default can be used for all 'TrainAlg' arguments that return an object with a predict method returning a list with a 'class' component (a factor) containing the classification results.
Strfolds	Boolean flag indicating if the folds should be stratified according to the original class proportions (default), or randomly generated from the whole training sample, ignoring class membership.
kfold	Number of training sample folds to be created in each replication.
CVrep	Number of replications to be performed.

prior	The prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
loo	A boolean flag indicating if a leave-one-out strategy should be employed. When set to "TRUE" overrides the kfold and CVrep arguments.
...	Further arguments to be passed to 'TrainAlg' and 'EvalAlg'.

Value

A three dimensional array with the number of tested observations, and estimated classification errors for each combination of fold and replication tried. The array dimensions are defined as follows:
The first dimension runs through the different fold-replication combinations.
The second dimension represents the classes.
The third dimension has two named levels representing respectively the number of observations tested ("Nk"), and the estimated classification errors ("Clerr").

Author(s)

A. Pedro Duarte Silva

See Also

lda, qda, IData

Examples

```
## Not run:

# Compare performance of linear and quadratic discriminant analysis with
#  Covariance cases C1 and c4 on the ChinaT data set by 5-fold cross-validation
#  replicated twice

# Create an Interval-Data object containing the intervals for 899 observations
# on the temperatures by quarter in 60 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8])

# Classical (configuration 1) Linear Discriminant Analysis

CVldaC1 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=lda,CovCase=1,kfold=5,CVrep=2)
summary(CVldaC1[,,"Clerr"])
glberrors <- apply(CVldaC1[,,"Nk"]*CVldaC1[,,"Clerr"],1,sum)/apply(CVldaC1[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")

# Linear Discriminant Analysis with covariance case 3

CVldaC4 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=lda,CovCase=3,kfold=5,CVrep=2)
summary(CVldaC4[,,"Clerr"])
glberrors <- apply(CVldaC4[,,"Nk"]*CVldaC4[,,"Clerr"],1,sum)/apply(CVldaC4[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")
```

```
# Classical (configuration 1) Quadratic Discriminant Analysis

CVqdaC1 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=qda,CovCase=1,kfold=5,CVrep=2)
summary(CVqdaC1[,,"Clerr"])
glberrors <- apply(CVqdaC1[,,"Nk"]*CVqdaC1[,,"Clerr"],1,sum)/apply(CVqdaC1[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")

# Quadratic Discriminant Analysis with covariance case 3

CVqdaC4 <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=qda,CovCase=3,kfold=5,CVrep=2)
summary(CVqdaC4[,,"Clerr"])
glberrors <- apply(CVqdaC4[,,"Nk"]*CVqdaC4[,,"Clerr"],1,sum)/apply(CVqdaC4[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glberrors),"\n")


## End(Not run)
```

`extmatrix-class` *Class "extmatrix"*

Description

"extmatrix" is a simple extension of the base matrix class, that that accepts NULL objects as members.

Extends

Class "matrix", directly.

`fasttle-methods` *Methods for Function fasttle in Package 'MAINT.Data'*

Description

Performs maximum trimmed likelihood estimation by the fasttle algorithm

Usage

```
fasttle(Idt,
    CovCase=1:4,
    SelCrit=c("BIC","AIC"),
    alpha=control@alpha,
    nsamp = control@nsamp,
    seed=control@seed,
    trace=control@trace,
```

```
    use.correction=control@use.correction,
    ncsteps=control@ncsteps,
    getalpha=control@getalpha,
    getkdblstar=control@getkdblstar,
    outlin=control@outlin,
    trialmethod=control@trialmethod,
    m=control@m,
    reweighted = control@reweighted,
    otpType=control@otpType,
    control=RobEstControl(), ...)
```

Arguments

Idt	An IData object representing interval-valued entities.
CovCase	Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit	The model selection criterion.
alpha	Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*Idt@NIVar observations are used for computing the trimmed likelihood. Allowed values are between 0.5 and 1.
nsamp	Number of subsets used for initial estimates. Note that when argument 'getalpha' is set to "TwoStep" the final value of 'alpha' is estimated by a two-step procedure and the value of argument 'alpha' is only used to specify the size of the samples used in the first step.
seed	Initial seed for random generator, like .Random.seed, see rrcov.control.
trace	Logical (or integer) indicating if intermediate results should be printed; defaults to FALSE.
use.correction	whether to use finite sample correction factors; defaults to TRUE.
ncsteps	The maximum number of concentration steps used each iteration of the fasttle algorithm.
getalpha	Argument specifying if the 'alpha' parameter (roughly the percentage of the sample used for computing the trimmed likelihood) should be estimated from the data, or if the value of the argument 'alpha' should be used instead. When set to "TwoStep", 'alpha' is estimated by a two-step procedure with the value of argument 'alpha' specifying the size of the samples used in the first step. Otherwise, with the value of argument 'alpha' is used directly.
getkdblstar	Argument specifying the size of the initial small (in order to minimize the probability of outliers) subsets. If set to the string "Twopplusone" (default) the initial sets have twice the number of interval-value variables plus one (i.e., they are the smaller samples that lead to a non-singular covariance estimate). Otherwise, an integer with the size of the initial sets.
outlin	The type of outliers to be considered. "MidPandLogR" if outliers may be present in both MidPpoints and LogRanges, "MidP" if outliers are only present in MidPpoints, or "LogR" if outliers are only present in LogRanges.

`trialmethod` The method to find a trial subset used to initialize each replication of the fasttle algorithm. The current options are "simple" (default) that simply selects 'kdblstar' observations at random, and "Poolm" that divides the original sample into 'm' non-overlaping subsets, applies the 'simple trial' and the refinement methods to each one of them, and merges the results into a trial subset.

`m` Number of non-overlaping subsets used by the trial method when the argument of 'trialmethod' is set to 'Poolm'.

`reweighted` Should a (Re)weighted estimate of the covariance matrix be used in the computation of the trimmed likelihood or just a "raw" covariance estimate; default is (Re)weighting.

`otpType` The amount of output returned by fasttle. Current options are "OnlyEst" (default) where only an 'IdtE' object with the fasttle estimates is returned, "SetMD2andEst" which returns a list with an 'IdtE' object of fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and "SetMD2EstandPrfSt" wich returns a list with the previous three components plust a list of some performance statistics concerning the algorithm execution.

`control` a list with estimation options - this includes those above provided in the function specification. See `RobEstControl` for the defaults. If `control` is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object.

`...` Further arguments to be passed to internal functions of `fasttle`.

Value

If argument 'otpType' is set to "OnlyEst", an object of class 'IdtE' with the fasttle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

If argument 'otpType' is set to "SetMD2andEst" a list with the following components:

`sol` An object of class 'IdtE' with the fasttle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

`Set` A vector with the final trimmed subset elements used to compute the fasttle estimates.

`RobMD2` A vector with the robust squared Mahalanobis distances used to select the trimmed subset.

If argument 'otpType' is set to "SetMD2EstandPrfSt" a list with the following components:

`sol` An object of class 'IdtE' with the fasttle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

`Set` A vector with the final trimmed subset elements used to compute the fasttle estimates.

`RobMD2` A vector with the robust squared Mahalanobis distances used to select the trimmed subset.

`PerfSt` A a list with the following components:
RepSteps: A list with one component by Covariance Configuration, containing a vector with the number of refinement steps performed by the fasttle algorithm by replication.
RepLogLik: A list with one component by Covariance Configuration, containing a vector with the best log-likelihood found be fasttle algorithm by replication.
StpLogLik: A list with one component by Covariance Configuration, containing a matrix with the evolution of the log-likelihoods found be fasttle algorithm by replication and refinement step.

Methods

`signature(Idt = "IData")` Performs maximum trimmed likelihood estimation for interval-valued data using the fasttle algorithm, assuming a Gaussian distribution, and considering alternative variance-covariance matrix configurations

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

`fulltle`

Examples

```
## Not run:
# Create an Interval-Data object containing the intervals of temperatures by quarter
# for 899 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8])

# Estimate parameters using the fast trimmed maximum likelihood estimator, and assuming that one of
# the C2, C3 or C4 restricted Covariance Cases holds

Chinafasttle <- fasttle(ChinaT,CovCase=2:4)
cat("China maximum trimmed likelihood estimation results =\n")
print(Chinafasttle)

## End(Not run)
```

`fulltle-methods` *Methods for Function fulltle in Package 'MAINT.Data'*

Description

Performs maximum trimmed likelihood estimation by an exact algorithm (full enumeratiom of all k-trimmed subsets)

Usage

```
fulltle(Idt, alpha=0.75, reweighted=TRUE, CorrF=c("smallsmp","consistent","none"),
  outlin=c("MidPandLogR","MidP","LogR"),
  CovCase=1:4, SelCrit=c("BIC","AIC"),
  force=FALSE,otpType=c("OnlyEst","SetMD2andEst"), ...)
```

Arguments

`Idt`	An IData object representing interval-valued entities.
`alpha`	Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*Idt@NIVar observations are used for computing the trimmed likelihood. Allowed values are between 0.5 and 1.
`reweighted`	Should a (Re)weighted estimate of the covariance matrix be used in the computation of the trimmed likelihood or just a "raw" covariance estimate; default is (Re)weighting.
`CorrF`	Whether to use sample correction factors for the covariance estimates; options are "smallsmp" (default), for small sample correction, "consistent", for consistency correction and "none" for no correction.
`outlin`	The type of outliers to be consideres. "MidPandLogR" if outliers may be present in both MidPpoints and LogRanges, "MidP" if outliers are only present in MidPpoints, or "LogR" if outliers are only present in LogRanges.
`CovCase`	Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
`SelCrit`	The model selection criterion.
`force`	A boolean flag indicating whether, for moderate or large data sets the algorithm should proceed anyway, regardless of an expected long excution time, due to exponential explosions in the number of different subsets that need to be avaluated by fulltle
`otpType`	The amount of output returned by fulltle. Current options are "OnlyEst" (default) where only an 'IdtE' object with the fulltle estimates is returned and "SetMD2andEst" which returns a list with an 'IdtE' object of fulltle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances.
`...`	Further arguments to be passed to internal functions of 'fulltle'.

Value

If argument 'otpType' is set to "OnlyEst", an object of class 'IdtE' with the fulltle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.

If argument 'otpType' is set to "SetMD2andEst" a list with the following components:

`sol`	An object of class 'IdtE' with the fulltle estimates, their log-likelihood values, and the value of the comparison criterion used to select the covariance configurations.
`Set`	A vector with the final trimmed subset elements used to compute the fulltle estimates.
`RobMD2`	A vector with the robust squared Mahalanobis distances used to select the trimmed subset.

Methods

`signature(Idt = "IData")` Performs maximum trimmed likelihood estimation for interval-valued data using an exact algorithm, and assuming a Gaussian distribution, and considering alternative variance-covariance matrix configurations

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

See Also

`fasttle`

Examples

```
# Create an Interval-Data object containing the intervals for characteristics
# of 27 cars models.

Cars <- IData(CarsData[1:8],Seq="MidPLogR_VarbyVar",
  VarNames=c("Price","EngineCapacity","TopSpeed","Acceleration"))

#Display the first and last observations

head(Cars)
tail(Cars)

## Not run:
# Estimate normal distributuion  parameters robustly by full maximum trimmed likelihood,
```

```
CarsTE <- fulltle(Cars)
cat("Cars data -- normal maximum trimmed likelhiood estimation results:\n")
print(CarsTE)

## End(Not run)
```

IData *Interval Data objects*

Description

'IData' creates IData objects from data frames of interval bounds or MidPoint/LogRange values of the interval-valued observations.

Usage

```
IData(Data,
Seq = c("LbUb_VarbyVar", "MidPLogR_VarbyVar", "AllLb_AllUb", "AllMidP_AllLogR"),
VarNames=NULL, ObsNames=row.names(Data))
```

Arguments

Data	a data frame of interval bounds or MidPoint/LogRange values.
Seq	the format of 'Data' data frame. Available options are: "LbUb_VarbyVar": lower bounds followed by upper bounds, variable by variable. "MidPLogR_VarbyVar": MidPoints followed by LogRanges, variable by variable. "AllLb_AllUb": all lower bounds followed by all upper bounds, in the same variable order. "AllMidP_AllLogR": all MidPoints followed all LogRanges, in the same variable order.
VarNames	An optional vector of names to be assigned to the Interval-Valued Variables.
ObsNames	An optional vector of names assigned to the individual observations.

Details

Objects of type 'IData', describe a data set of 'NObs' observations on 'NIVar' Interval-Valued variables. This function creates an interval-data object from a data-frame with either the lower and upper bounds of the observed intervals or by their midpoints and log-ranges.

See Also

IData

Examples

```
ChinaT <- IData(ChinaTemp[1:8],VarNames=c("T1","T2","T3","T4"))
cat("Summary of the ChinaT IData object:\n")  ; print(summary(ChinaT))
cat("ChinaT first ant last observations:\n")
print(head(ChinaT,n=3))
cat("\n...\n")
print(tail(ChinaT,n=3))
```

IData-class *Class "IData"*

Description

A data-array of interval-valued data is an array where each of the NObs rows, corresponding to each entity under analysis, contains the observed intervals of the NIVar descriptive variables.

Slots

`MidP`**:** A data-frame of the midpoints of the observed intervals

`LogR`**:** A data-frame of the logarithms of the ranges of the observed intervals

`ObsNames`**:** An optional vector of names assigned to the individual observations.

`VarNames`**:** An optional vector of names to be assigned to the Interval-Valued Variables.

`NObs`**:** Number of entities under analysis (cases)

`NIVar`**:** Number of interval variables

Methods

head `signature(x = "IData")`: head S4 method for the IData-class.

show `signature(object = "IData")`: show S4 method for the IData-class.

nrow `signature(x = "IData")`: returns the number of entities (observations).

ncol `signature(x = "IData")`: returns the number of Iterval Variables.

show `signature(object = "IData")`: show S4 method for the IData-class.

tail `signature(x = "IData")`: tail S4 method for the IData-class.

mle `signature(x = "IData")`: Maximum likelihood estimation.

fasttle `signature(x = "IData")`: Fast trimmed maximum likelihood estimation.

fulltle `signature(x = "IData")`: Exact trimmed maximum likelihood estimation.

mle `signature(x = "IData")`: Maximum likelihood estimation.

RobMxtDEst `signature(x = "IData")`: Robust estimation of distribution mixtures for interval-valued data.

MANOVA `signature(x = "IData")`: MANOVA tests on the interval-valued data.

lda `signature(x = "IData")`: Linear Discriminant Analysis using maximum likelihood parameter estimates of Gaussian mixtures.

qda `signature(x = "IData")`: Quadratic Discriminant Analysis using maximum likelihood parameter estimates of Gaussian mixtures.

Roblda `signature(x = "IData")`: Linear Discriminant Analysis using robust estimates of location and scatter.

Robqda `signature(x = "IData")`: Quadratic Discriminant Analysis using robust estimates of location and scatter.

snda `signature(x = "IData")`: Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Noirhomme-Fraiture, M., Brito, P. (2011), Far Beyond the Classical Data Models: Symbolic Data Analysis. *Statistical Analysis and Data Mining* **4**(2), 157–170.

See Also

`IData`, `mle`, `fasttle`, `fulltle`, `RobMxtDEst`, `MANOVA`, `lda`, `qda`, `Roblda`, `Robqda`

IdtE-class *Class "IdtE"*

Description

"IdtE" contains the results of model estimation. "IdtSngDE" extends "IdtE" assuming that the data can be carcherizes by a unique distribution (for instances, not considering partitions into different groups).

Slots

`ModelNames`**:** The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)

`ModelType`**:** Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented

`ModelConfig`**:** Configuration of the variance-covariance matrix: Case 1 through Case 4

`NIVar`: Number of interval variables

`SelCrit`: The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: The logarithms of the likelihood function for the different cases

`AICs`: Value of the AIC criterion

`BICs`: Value of the BIC criterion

`BestModel`: Bestmodel indicates the best model according to the chosen selection criterion

`SngD`: Boolean flag indicating whether a single or a mixture of distribution were estimated

Methods

BestModel `signature(Idt = "IdtE")`: Selects the best model according to the chosen selection criterion (currently, AIC or BIC)

show `signature(object = "IdtE")`: show S4 method for the IDtE-class

testMod `signature(Idt = "IdtE")`: Performs statistical likelihood-ratio tests that evaluate the goodness-of-fit of a nested model against a more general one.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`mle`, `fasttle`, `fulltle`, `MANOVA`, `RobMxtDEst`, `IData`

`Idtlda-class` *Class "Idtlda"*

Description

Idtlda contains the results of Linear Discriminant Analysis for the interval data

Slots

`prior`: Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.

`means`: Matrix with the mean vectors for each group

`scaling`: Matrix which transforms observations to discriminant functions, normalized so that within groups covariance matrix is spherical.

`N`: Number of observations

`CovCase`: Configuration case of the variance-covariance matrix: Case 1 through Case 4

Methods

predict `signature(object = "Idtlda")`: Classifies interval-valued observations in conjunction with lda.

show `signature(object = "Idtlda")`: show S4 method for the IDdtlda-class

CovCase `signature(object = "Idtlda")`: Returns the configuration case of the variance-covariance matrix

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

`qda`, `MANOVA`, `Roblda`, `Robqda`, `IData`

IdtMANOVA-class *Class "IdtMANOVA"*

Description

IdtMANOVA extends`"LRTest"` directly, containing the results of MANOVA tests on the interval-valued data. This class is not used directly, but is the basis for different specializations according to the model assumed for the distribution in each group. In paticular, the following specializations of IdtMANOVA are currently implemented:

"IdtClMANOVA" extends IdtMANOVA, assuming a classical (i.e., homocedastic gaussian) setup.

"IdtHetNMANOVA" extends IdtMANOVA, assuming a heterocedastic gaussian set-up.

"IdtLocNMANOVA" extends IdtMANOVA, assuming a Skew-Normal location model set-up.

"IdtLocNSNMANOVA" extends IdtMANOVA, assuming either a homocedastic gaussian or Skew-Normal location model set-up.

"IdtGenNMANOVA" extends IdtMANOVA, assuming a Skew-Normal general model set-up.

"IdtLocNSNMANOVA" extends IdtMANOVA, assuming either a heterocedastic gaussian or Skew-Normal general model set-up.

Slots

`NIVar`: Number of interval variables.

`grouping`: Factor indicating the group to which each observation belongs to.

`H0res`: Model estimates under the null hypothesis.

`H1res`: Model estimates under the alternative hypothesis.

`QuiSq`: Inherited from class `"LRTest"`. Value of the Qui-Square statistics corresponding to the performed test.

`df`: Inherited from class `"LRTest"`. Degrees of freedom of the Qui-Square statistics.

`pvalue`: Inherited from class `"LRTest"`. p-value of the Qui-Square statistics value, obtained from the Qui-Square distribution with df degrees of freedom.

`H0logLik`: Inherited from class `"LRTest"`. Logarithm of the Likelihood function under the null hypothesis.

`H1logLik`: Inherited from class `"LRTest"`. Logarithm of the Likelihood function under the alternative hypothesis.

Methods

show `signature(object = "IdtMANOVA")`: show S4 method for the IdtMANOVA-classes.

H0res `signature(object = "IdtMANOVA")`: retrieves the model estimates under the null hypothesis.

H1res `signature(object = "IdtMANOVA")`: retrieves the model estimates under the alternative hypothesis.

lda `signature(x = "IdtClMANOVA")`: Linear Discriminant Analysis using the estimated model parameters.

lda `signature(x = "IdtLocNSNMANOVA")`: Linear Discriminant Analysis using the estimated model parameters.

qda `signature(x = "IdtHetNMANOVA")`: Quadratic Discriminant Analysis using the estimated model parameters.

qda `signature(x = "IdtGenNSNMANOVA")`: Quadratic Discriminant Analysis using the estimated model parameters.

snda `signature(x = "IdtLocSNMANOVA")`: Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a "location" model (i.e., groups differ only in location parameters).

snda `signature(x = "IdtLocNSNMANOVA")`: Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a "location" model (i.e., groups differ only in location parameters).

snda `signature(x = "IdtGenSNMANOVA")`: Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a general model (i.e., groups differ in all parameters).

snda `signature(x = "IdtGenNSNMANOVA")`: Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a general model (i.e., groups differ in all parameters).

Extends

Class "LRTest", directly.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012): "Modelling Interval Data with Normal and Skew-Normal Distributions". Journal of Applied Statistics, Volume 39, Issue 1, 3-20.

See Also

MANOVA, lda, qda, snda, Roblda, Robqda, RobMxtDEst, IData

IdtMxE-class *Class "IdtMxE"*

Description

IdtMxE contains the results of a mixture model estimation.

Slots

grouping: Factor indicating the group to which each observation belongs to

ModelNames: Inherited from class "IdtE". The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)

ModelType: Inherited from class "IdtE". Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented.

ModelConfig: Inherited from class "IdtE". Configuration of the variance-covariance matrix: Case 1 through Case 4

NIVar: Inherited from class "IdtE". Number of interval variables

SelCrit: Inherited from class "IdtE". The model selection criterion; currently, AIC and BIC are implemented

logLiks: Inherited from class "IdtE". The logarithms of the likelihood function for the different cases

AICs: Inherited from class "IdtE". Value of the AIC criterion

BICs: Inherited from class "IdtE". Value of the BIC criterion

BestModel: Inherited from class "IdtE". Bestmodel indicates the best model according to the chosen selection criterion

SngD: Inherited from class "IdtE". Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class "IdtMxE"

Ngrps: Number of mixture components

Extends

Class "IdtE", directly.

Methods

No methods defined with class "IdtMxE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

IdtE, IData, MANOVA, RobMxtDEst

IdtMxNandSNDE-class *Class "IdtMxNandSNDE"*

Description

IdtMxNandSNDE contains the results of a mixture model estimation; Normal an Skew-Normal models are considered, with the four different possible variance-covariance configurations.

Slots

NMod: Estimates of the mixture model for the Gaussian case

SNMod: Estimates of the mixture model for the Skew-Normal case

grouping: Inherited from class "IdtMxE". Factor indicating the group to which each observation belongs to

ModelNames: Inherited from class "IdtE". The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)

ModelType: Inherited from class "IdtE". Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented

ModelConfig: Inherited from class "IdtE". Configuration case of the variance-covariance matrix: Case 1 through Case 4

NIVar: Inherited from class "IdtE". Number of interval variables

SelCrit: Inherited from class "IdtE". The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class `"IdtMxNandSNDE"`

`Ngrps`: Inherited from class `"IdtMxE"`. Number of mixture components

Extends

Class `"IdtMxE"`, directly. Class `"IdtE"`, by class "IdtMxE", distance 2.

Methods

No methods defined with class "IdtMxNandSNDE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`MANOVA`, `RobMxtDEst`, `IData`

`IdtMxNDE-class` *Class "IdtMxNDE"*

Description

"IdtMxNDE" contains the results of a mixture Normal model maximum likelihood parameter estimation, with the four different possible variance-covariance configurations.
"IdtMxtNDE" is an union of classes "IdtMxNDE" and "IdtMxNDRE", the later one containing the results of mixture Normal model parameter estimation by robust methods.

Slots

`Hmcdt`: Indicates whether we consider an homocedastic (TRUE) or a hetereocedasic model (FALSE)

`mleNmuE`: Matrix with the maximum likelihood mean vectors estimates by group (each row refers to a group)

`mleNmuEse`: Matrix with the maximum likelihood means' standard errors by group (each row refers to a group)

`CovConfCases`: List of the considered configurations

`grouping`: Inherited from class `"IdtMxE"`. Factor indicating the group to which each observation belongs to

`ModelNames`: Inherited from class `"IdtE"`. The model acronym formed by a "N", indicating a Normal model, followed by the configuration (Case 1 through Case 4)

`ModelType`: Inherited from class `"IdtE"`. Indicates the model; always set to "Normal" in objects of the IdtMxNDE class

`ModelConfig`: Inherited from class `"IdtE"`. Configuration case of the variance-covariance matrix: Case 1 through Case 4

`NIVar`: Inherited from class `"IdtE"`. Number of interval variables

`SelCrit`: Inherited from class `"IdtE"`. The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class `"IdtMxNDE"`

`Ngrps`: Inherited from class `"IdtMxE"`. Number of mixture components

Extends

Class `"IdtMxE"`, directly. Class `"IdtNDE"`, directly. Class `"IdtE"`, by class "IdtMxE", distance 2.

Methods

lda `signature(x = "IdtMxtNDE")`: Linear Discriminant Analysis using the estimated model parameters.

qda `signature(x = "IdtMxtNDE")`: Quadratic Discriminant Analysis using the estimated model parameters.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

MANOVA, IData, IdtMxNDRE

`IdtMxNDRE-class` *Class "IdtMxNDE"*

Description

"IdtMxNDRE" contains the results of a mixture Normal model robust parameter estimation, with the four different possible variance-covariance configurations.

Slots

`Hmcdt`: Indicates whether we consider an homocedastic (TRUE) or a hetereocedasic model (FALSE)

`RobNmuE`: Matrix with the robust mean vectors estimates by group (each row refers to a group)

`CovConfCases`: List of the considered configurations

`grouping`: Inherited from class `"IdtMxE"`. Factor indicating the group to which each observation belongs to

`ModelNames`: Inherited from class `"IdtE"`. The model acronym formed by a "N", indicating a Normal model, followed by the configuration (Case 1 through Case 4)

`ModelType`: Inherited from class `"IdtE"`. Indicates the model; always set to "Normal" in objects of the IdtMxNDRE class

`ModelConfig`: Inherited from class `"IdtE"`. Configuration case of the variance-covariance matrix: Case 1 through Case 4

`NIVar`: Inherited from class `"IdtE"`. Number of interval variables

`SelCrit`: Inherited from class `"IdtE"`. The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class `"IdtMxNDRE"`

`Ngrps`: Inherited from class `"IdtMxE"`. Number of mixture components

Extends

Class "IdtMxE", directly. Class "IdtNDE", directly. Class "IdtE", by class "IdtMxE", distance 2.

Methods

No methods defined with class "IdtMxNDRE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

See Also

RobMxtDEst, IData, IdtMxNDE, IdtMxtNDE

IdtMxSNDE-class *Class "IdtMxSNDE"*

Description

IdtMxSNDE contains the results of a mixture model estimation for the Skew-Normal model, with the four different possible variance-covariance configurations.

Slots

Hmcdt: Indicates whether we consider an homocedastic location model (TRUE) or a general model (FALSE)

CovConfCases: List of the considered configurations

grouping: Inherited from class "IdtMxE". Factor indicating the group to which each observation belongs to

ModelNames: Inherited from class "IdtE". The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)

ModelType: Inherited from class "IdtE". Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented

ModelConfig: Inherited from class "IdtE". Configuration case of the variance-covariance matrix: Case 1 through Case 4

`NIVar`: Inherited from class `"IdtE"`. Number of interval variables

`SelCrit`: Inherited from class `"IdtE"`. The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to FALSE in objects of class `"IdtMxSNDE"`

`Ngrps`: Inherited from class `"IdtMxE"`. Number of mixture components

Extends

Class `"IdtMxE"`, directly. Class `"IdtSNDE"`, directly. Class `"IdtE"`, by class "IdtMxE", distance 2.

Methods

No methods defined with class "IdtMxSNDE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`MANOVA`, `IData`

IdtNandSNDE-class *Class "IdtNandSNDE"*

Description

IdtNandSNDE is a class union used for storing the estimation results of Normal and Skew-Normal modelizations for Interval Data.

Methods

coef `signature(coef = "IdtNandSNDE")`: extracts parameter estimates from objects of type IdtNandSNDE.

stdEr `signature(x = "IdtNandSNDE")`: extracts standard errors from objects of type IdtNandSNDE.

vcov `signature(x = "IdtNandSNDE")`: extracts an estimate of the variance-covariance matrix of the parameters estimators for objects of type IdtNandSNDE.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

IData, mle, fasttle, fulltle, MANOVA, RobMxtDEst, IdtSngNandSNDE, IdtMxNandSNDE

IdtNDE-class *Class "IdtNDE"*

Description

IdtNDE is a class union used for storing the estimation results of Normal modelizations for Interval Data.

Methods

coef `signature(coef = "IdtNDE")`: extracts parameter estimates from objects of type IdtNDE

stdEr `signature(x = "IdtNDE")`: extracts standard errors from objects of type IdtNDE

vcov `signature(x = "IdtNDE")`: extracts an estimate of the variance-covariance matrix of the paramenters estimators for objects of type IdtNDE

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

IData, mle, fasttle, fulltle, MANOVA, RobMxtDEst, IdtSngNDE, IdtMxNDE

`Idtqda-class` *Class "Idtqda"*

Description

Idtqda contains the results of Quadratic Discriminant Analysis for the interval data

Slots

`prior`**:** Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.

`means`**:** Matrix with the mean vectors for each group

`scaling`**:** A three-dimensional array. For each group, g, scaling[,,g] is a matrix which transforms interval-valued observations so that within-groups covariance matrix is spherical.

`ldet`**:** Vector of half log determinants of the dispersion matrix.

`lev`**:** Levels of the grouping factor

`CovCase`**:** Configuration case of the variance-covariance matrix: Case 1 through Case 4

Methods

predict `signature(object = "Idtqda")`: Classifies interval-valued observations in conjunction with qda.

show `signature(object = "Idtqda")`: show S4 method for the Idtqda-class

CovCase `signature(object = "Idtqda")`: Returns the configuration case of the variance-covariance matrix

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

`qda`, `MANOVA`, `Robqda`, `IData`

IdtSNDE-class *Class "IdtSNDE"*

Description

IdtSNDE is a class union used for storing the estimation results of Skew-Normal modelizations for Interval Data.

Methods

coef `signature(coef = "IdtSNDE")`: extracts parameter estimates from objects of type IdtSNDE.

stdEr `signature(x = "IdtSNDE")`: extracts standard errors from objects of type IdtSNDE.

vcov `signature(x = "IdtSNDE")`: extracts an asymptotic estimate of the variance-covariance matrix of the paramenters estimators for objects of type IdtNDE.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`IData`, `mle`, `MANOVA`, `IdtSngSNDE`, `IdtMxSNDE`

IdtSNgenda-class *Class "IdtSNgenda"*

Description

IdtSNgenda contains the results of discriminant analysis for the interval data, based on a general Skew-Normal model.

Slots

`prior`**:** Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.

`ksi`**:** Matrix with the direct location parameter ("ksi") estimates for each group.

`eta`**:** Matrix with the direct scaled skweness parameter ("eta") estimates for each group.

`scaling`**:** For each group g, scaling[,,g] is a matrix which transforms interval-valued observations so that in each group the scale-association matrix ("Omega") is spherical.

`mu`**:** Matrix with the centred location parameter ("mu") estimates for each group.

`gamma1`**:** Matrix with the centred skweness parameter ("gamma1") estimates for each group.

`ldet`**:** Vector of half log determinants of the dispersion matrix.

`lev`**:** Levels of the grouping factor.

`CovCase`**:** Configuration case of the variance-covariance matrix: Case 1 through Case 4

Methods

predict `signature(object = "IdtSNgenda")`: Classifies interval-valued observations in conjunction with snda.

show `signature(object = "IdtSNgenda")`: show S4 method for the IdtSNgenda-class

CovCase `signature(object = "IdtSNgenda")`: Returns the configuration case of the variance-covariance matrix

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

`MANOVA`, `snda`, `IData`

`IdtSngNandSNDE-class` *Class "IdtSngNandSNDE"*

Description

IdtSngNandSNDE contains the results of a single class model estimation for the Normal and the Skew-Normal distributions, with the four different possible variance-covariance configurations.

Slots

`NMod`**:** Estimates of the single class model for the Gaussian case

`SNMod`**:** Estimates of the single class model for the Skew-Normal case

`ModelNames`**:** Inherited from class `"IdtE"`. The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration (Case 1 through Case 4)

`ModelType`**:** Inherited from class `"IdtE"`. Indicates the model; currently, Gaussian or Skew-Normal distributions are implemented

`ModelConfig`**:** Inherited from class `"IdtE"`. Configuration of the variance-covariance matrix: Case 1 through Case 4

`NIVar`**:** Inherited from class `"IdtE"`. Number of interval variables

`SelCrit`: Inherited from class `"IdtE"`. The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Bestmodel indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to TRUE in objects of class `"IdtSngNandSNDE"`

Extends

Class `"IdtSngDE"`, directly. Class `"IdtE"`, by class "IdtSngDE", distance 2.

Methods

No methods defined with class "IdtSngNandSNDE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`IData`, `mle`, `fasttle`, `fulltle`

`IdtSngNDE-class` *Class "IdtSngNDE"*

Description

Contains the results of a single class maximum likelihood estimation for the Normal distribution, with the four different possible variance-covariance configurations.

Slots

`mleNmuE`: Matrix with the maximum likelihood mean vectors estimates

`mleNmuEse`: Matrix with the maximum likelihood means' standard errors by group

`CovConfCases`: List of the considered configurations

`ModelNames`: Inherited from class `"IdtE"`. The model acronym formed by a "N", indicating a Normal model, followed by the configuration (Case 1 through Case 4)

`ModelType`: Inherited from class `"IdtE"`. Indicates the model; always set to "Normal" in objects of the "IdtSngNDE" class

`ModelConfig`: Inherited from class `"IdtE"`. Configuration of the variance-covariance matrix: Case 1 through Case 4

`NIVar`: Inherited from class `"IdtE"`. Number of interval variables

`SelCrit`: Inherited from class `"IdtE"`. The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Bestmodel indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to TRUE in objects of class `"IdtSngNDE"`

Extends

Class `"IdtSngDE"`, directly. Class `"IdtE"`, by class "IdtSngDE", distance 2.

Methods

No methods defined with class "IdtSngNDE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

`IData`, `mle`, `IdtSngNDE`

IdtSngNDRE-class *Class "IdtSngNDRE"*

Description

Contains the results of a single class robust estimation for the Normal distribution, with the four different possible variance-covariance configurations.

Slots

`RobNmuE`: Matrix with the maximum likelihood mean vectors estimates

`CovConfCases`: List of the considered configurations

`ModelNames`: Inherited from class `"IdtE"`. The model acronym formed by a "N", indicating a Normal model, followed by the configuration (Case 1 through Case 4)

`ModelType`: Inherited from class `"IdtE"`. Indicates the model; always set to "Normal" in objects of the "IdtSngNDRE" class

`ModelConfig`: Inherited from class `"IdtE"`. Configuration of the variance-covariance matrix: Case 1 through Case 4

`NIVar`: Inherited from class `"IdtE"`. Number of interval variables

`SelCrit`: Inherited from class `"IdtE"`. The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Bestmodel indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to TRUE in objects of class `"IdtSngNDRE"`

Extends

Class `"IdtSngDE"`, directly. Class `"IdtE"`, by class "IdtSngDE", distance 2.

Methods

No methods defined with class "IdtSngNDRE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

See Also

IData, fasttle, fulltle, IdtSngNDE

`IdtSngSNDE-class` *Class "IdtSngSNDE"*

Description

Contains the results of a single class maximul likelihood estimation for the Skew-Normal distribution, with the four different possible variance-covariance configurations.

Slots

`CovConfCases`: List of the considered configurations

`ModelNames`: The model acronym, indicating the model type (currently, N for Normal and SN for Skew-Normal), and the configuration Case (C1 to C4) for the covariance matrix

`ModelNames`: Inherited from class `"IdtE"`. The model acronym formed by a "SN", indicating a skew-Normal model, followed by the configuration (Case 1 through Case 4)

`ModelType`: Inherited from class `"IdtE"`. Indicates the model; always set to "SkewNormal" in objects of the IdtSngSNDE class

`ModelConfig`: Inherited from class `"IdtE"`. Configuration case of the variance-covariance matrix: Case 1 through Case 4

`NIVar`: Inherited from class `"IdtE"`. Number of interval variables

`SelCrit`: Inherited from class `"IdtE"`. The model selection criterion; currently, AIC and BIC are implemented

`logLiks`: Inherited from class `"IdtE"`. The logarithms of the likelihood function for the different cases

`AICs`: Inherited from class `"IdtE"`. Value of the AIC criterion

`BICs`: Inherited from class `"IdtE"`. Value of the BIC criterion

`BestModel`: Inherited from class `"IdtE"`. Indicates the best model according to the chosen selection criterion

`SngD`: Inherited from class `"IdtE"`. Boolean flag indicating whether a single or a mixture of distribution were estimated. Always set to TRUE in objects of class `"IdtSngSNDE"`

Extends

Class "IdtSngDE", directly. Class "IdtE", by class "IdtSngDE", distance 2.

Methods

No methods defined with class "IdtSngSNDE" in the signature.

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Azzalini, A. and Dalla Valle, A. (1996), The multivariate skew-normal distribution. *Biometrika* **83**(4), 715–726.

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

See Also

mle, IData, IdtSngNDE

IdtSNlocda-class *Class "IdtSNlocda"*

Description

IdtSNlocda contains the results of Discriminant Analysis for the interval data, based on a location Skew-Normal model.

Slots

prior: Prior probabilities of class membership; if unspecified, the class proportions for the training set are used; if present, the probabilities should be specified in the order of the factor levels.

ksi: Matrix with the direct location parameter ("ksi") estimates for each group.

eta: Vector with the direct scaled skweness parameter ("eta") estimates.

scaling: Matrix which transforms observations to discriminant functions, normalized so that the within groups scale-association matrix ("Omega") is spherical.

mu: Matrix with the centred location parameter ("mu") estimates for each group.

gamma1: Vector with the centred skweness parameter ("gamma1") estimates.

N: Number of observations.

CovCase: Configuration case of the variance-covariance matrix: Case 1 through Case 4

Methods

predict `signature(object = "IdtSNlocda")`: Classifies interval-valued observations in conjunction with snda.

show `signature(object = "IdtSNlocda")`: show S4 method for the IDdtlda-class

CovCase `signature(object = "IdtSNlocda")`: Returns the configuration case of the variance-covariance matrix

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Duarte Silva, A.P. and Brito, P. (2015), Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification* **39**(3), 516–541.

See Also

`snda`, `MANOVA`, `IData`

`LRTest-class` *Class "LRTest"*

Description

LRTest contains the results of likelihood ratio tests

Slots

`QuiSq`**:** Value of the Qui-Square statistics corresponding to the performed test

`df`**:** Degrees of freedom of the Qui-Square statistics

`pvalue`**:** p-value of the Qui-Square statistics value, obtained from the Qui-Square distribution with df degrees of freedom

`H0logLik`**:** Logarithm of the Likelihood function under the null hypothesis

`H1logLik`**:** Logarithm of the Likelihood function under the alternative hypothesis

Methods

show `signature(object = "LRTest")`: show S4 method for the LRTest-class

Author(s)

Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>

See Also

mle, IData, ConfTests, MANOVA

MANOVA-methods *Methods for Function MANOVA in Package 'MAINT.Data'*

Description

Function MANOVA performs MANOVA tests based on likelihood ratios allowing for both Gaussian and Skew-Normal distribtuions and homocedastic or heterocedastic setups. Methods H0res and H1res retrieve the model estimates under the null and alternative hypothesis, and method show displays the MANOVA results.

Usage

```
MANOVA(Idt, grouping, Model=c("Normal","SKNormal","NrmandSKN"), CovCase=1:4,
SelCrit=c("BIC","AIC"), Mxt=c("Hom","Het","Loc","Gen"), CVtol=1.0e-5,
OptCntrl=list(), onerror=c("stop","warning","silentNull"), ...)

## S4 method for signature 'IdtMANOVA'
H0res(object)
## S4 method for signature 'IdtMANOVA'
H1res(object)
## S4 method for signature 'IdtMANOVA'
show(object)
```

Arguments

object	An object representing a MANOVA analysis on interval-valued entities.
Idt	An IData object representing interval-valued entities.
grouping	Factor indicating the group to which each observation belongs to.
Model	The joint distribution assumed for the MidPoint and LogRanges. Current alternatives are "Normal" for Gaussian, distributions, "SKNormal" for Skew-Normal and "NrmandSKN" for both Gaussian and Skew-Normal distributions.
CovCase	Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit	The model selection criterion.

Mxt	Indicates the type of mixing distributions to be considered. Current alternatives are "Hom" (homocedastic) and "Het" (hetereocedasic) for Gaussian models, "Loc" (location model – groups differ only on their location parameters) and "Gen" "Loc" (general model – groups differ on all parameters) for Skew-Normal models.
CVtol	Tolerance level for absolute value of the coefficient of variation of non-constant variables. When a MidPoint or LogRange has an absolute value within-groups coefficient of variation below CVtol, it is considered to be a constant.
OptCntrl	List of optional control parameters to passed to the optimization routine. See the documentation of RepLOptim for a description of the available options.
onerror	Indicates whether an error in the optimization algorithm should stop the current call, generate a warning, or return silently a NULL object.
...	Other named arguments.

Value

An object of type IdtMANOVA, containing the estimation and test results.

See Also

IdtMANOVA, RepLOptim

Examples

```
#Create an Interval-Data object containing the intervals of temperatures by quarter
# for 899 Chinese meteorological stations.
ChinaT <- IData(ChinaTemp[1:8])

#MANOVA tests assuming that one of C2, C3 or C4 restricted configuration cases hold

#Classical (homocedastic) MANOVA tests

ManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,CovCase=2:4)
cat("China, MANOVA by geografical regions results =\n")
print(ManvChina)

#Heterocedastic MANOVA tests

HetManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,Mxt="Het",CovCase=2:4)
cat("China, heterocedastic MANOVA by geografical regions results =\n")
print(HetManvChina)

#Skew-Normal based MANOVA assuming the the groups differ only according to location parameters
## Not run:

SKNLocManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,Model="SKNormal",Mxt="Loc",CovCase=2:4)
cat("China, Skew-Normal MANOVA (location model) by geografical regions results =\n")
print(SKNLocManvChina)
```

```
#Skew-Normal based MANOVA assuming the the groups may differ in all parameters

SKNGenManvChina <- MANOVA(ChinaT,ChinaTemp$GeoReg,Model="SKNormal",Mxt="Gen",CovCase=2:4)
cat("China, Skew-Normal MANOVA (general model)  by geografical regions results =\n")
print(SKNGenManvChina)


## End(Not run)
```

---

mle-methods *Methods for Function mle in Package 'MAINT.Data'*

Description

Performs maximum likelihood estimation for parametric models of interval data

Usage

```
## S4 method for signature 'IData'
mle(Idt, Model=c("Normal","SKNormal","NrmandSKN"), CovCase=1:4,
  SelCrit=c("BIC","AIC"), OptCntrl=list(), ...)
```

Arguments

Idt	An IData object representing interval-valued entities.
Model	The joint distribution assumed for the MidPoint and LogRanges. Current alternatives are "Normal" for Gaussian, distributions, "SNNormal" for Skew-Normal and "NrmandSKN" for both Gaussian and Skew-Normal distributions.
CovCase	Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit	The model selection criterion.
OptCntrl	List of optional control parameters to passed to the optimization routine. See the documentation of RepLOptim for a description of the available options.
...	Other named arguments.

References

Brito, P., Duarte Silva, A. P. (2012): "Modelling Interval Data with Normal and Skew-Normal Distributions". Journal of Applied Statistics, Volume 39, Issue 1, 3-20.

See Also

IData, RepLOptim

Examples

```
# Create an Interval-Data object containing the intervals of temperatures by quarter
# for 899 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8])

# Estimate parameters by maximum likelihood, assuming that one of
# the C2, C3 or C4 restricted Covariance configurations holds

ChinaE <- mle(ChinaT,CovCase=2:4)
cat("China maximum likelhiood estimation results =\n")
print(ChinaE)
cat("Standard Errors of Estimators:\n")
print(stdEr(ChinaE))
```

RepLOptim *Repeated Local Optimization*

Description

'RepLOptim' Tries to minimize a function calling local optimizers several times from different random starting points.

Usage

```
RepLOptim(start, parsd, fr, gr=NULL, inphess=NULL, ..., method="nlminb",
 lower=NULL, upper=NULL, rethess=FALSE, parmstder=FALSE, control=list())
```

Arguments

start	Vector of starting points used in the first call of the local optimizer.
parsd	Vector of standard deviations for the parameter distribution generating starting points for the local optimizer.
fr	The function to be minimized. If method is neither "nlminb" or "L-BFGS-B", fr should accept a lbound and an ubound arguments for the parameter bounds, and should enforce these bounds before calling the local optimization routine.
gr	A function to return the gradient for the "nlminb", "BFGS", "CG" and L-BFGS-B methods. If it is 'NULL', a finite-difference approximation will be used. For the "SANN" method it specifies a function to generate a new candidate point. If it is 'NULL' a default Gaussian Markov kernel is used.
inphess	A function to return the hessian for the "nlminb" method. Must return a square matrix of order 'length(parmean)' with the different hessian elements in its lower triangle. It is ignored if method component of the control list is not set to its "nlminb" default.
...	Further arguments to be passed to 'fr', 'gr' and 'inphess'.
method	The method to be used. See 'Details'.

lower	Vector of parameter lower bounds. Set to '-Inf' (no bounds) by default.
upper	Vector of parameter upper bounds. Set to 'Inf' (no bounds) by default.
rethess	Bolean flag indicating wether a numerically evaluated hessian matrix at the optimum should be computed and returned. Not available for the "nlminb" method.
parmstder	Bolean flag indicating wether parameter assymptotic standard errors based on the inverse hessian approximation to the Fisher information matrix should be computed and returned. Only available if hessian is set to TRUE and if a local miminum with a positive-definite hessian was indeed found. This requirement may fail if 'nrep' and 'niter' (and maybe 'neval') are not large enough, and for non-trivial problems of moderate or high dimensionality may never be satisfied because of numerical difficulties.
control	A list of control parameters. See below for details.

Details

'RepLOptim' Tries to minimize a function by calling local optimizers several times from different starting points. The starting point used in the first call the the local optimizer is the value of the argument 'start'. Subsquent calls use starting points generated from uniform distributions of independent variates with means equal to the current best parameter values and standard deviations equal to the values of the argument 'parsd'. If parameter bounds are specified and the uniform limits implied by 'parsd' violate those bounds, these limits are replaced by the corresponding bounds.

The choice of the local optimizer is made by value of the 'method' argument. This argument can be a function object implementing the optimizer or a string describing an available R method. In the latter case current alternatives are: "nlminb" (default) for the 'nlminb' port routine, "nlm" for the 'nlm' function and "Nelder-Mead", "L-BFGS-B", "CG", "L-BFGS-B" and "SANN" for the corresponding methods of the 'optim' function.

Arguments for controling the behaviour of the local optimizer can be specified as components of `control` list. This list can include any of the following components:

`maxrepet` Maximum time of repetions of the same minimum objective value, before RepLOptim is stoped and the current best solution is returned. By default set to 2.

`maxnoimprov` Maximum number of times the local optimizer is called without improvements in the minimum objective value, before RepLOptim is stoped and the current best solution is returned. By default set to 50.

`maxreplic` Maximum number of times the local optimizer is called and returns a valid solution before RepLOptim is stoped and the current best solution is returned. By default set to 250.

`allrep` Total maximum number of replications (including those leading to non-valid solutions) performed. By default equals ten times the value of maxreplic. Ignored when objbnd is set to 'Inf'.

`maxiter` Maximum number of iterations performed in each call to the local optimizer. By default set to 500 except with the "SANN" mehtod, when by default is set to 1500.

`maxeval` Maximum number of function evaluations (nlminb method only) performed in each call to the nlminb optimizer. By defaults set to 1000.

`RLOtol` The relative convergence tolerance of the local optimizer. The local optimizer stops if it is unable to reduce the value by a factor of 'RLOtol *(abs(val) + reltol)' at a step. Ignored when method is set to "nlm". By default set to the square root of the computer precision, i.e. to 'sqrt(.Machine$double.eps)'.

`HesEgtol` Numerical tolerance used to ensure that the hessian is non-singular. If the last eigenvalue of the hessian is positive but the ratio between it and the first eigenvalue is below HesEgtol the hessian is considered to be semi-definite and the parameter assymptotic standard errors are not computed. By default set to the square root of the computer precision, i.e. to 'sqrt(.Machine$double.eps)'.

`objbnd` Upper bound for the objective. Only solutions leading to objective values below objbnd are considered as valid.

Value

A list with the following components:

`par`	The best result found for the parameter vector.
`val`	The best value (minimum) found for the function fr.
`vallist`	A vector with the best values found for each starting point.
`iterations`	Number the iterations performed by the local optimizer in the call that generated the best result.
`vallis`	A vector with the best values found for each starting point.
`counts`	number of times the function fr was evaluated in the call that generated the result returned.
`convergence`	Code with the convergence status returned by the local optimizer.
`message`	Message generated by the local optimizer.
`hessian`	Numerically evaluated hessian of fr at the result returned. Only returned when the parameter hessian is set to TRUE.
`hessegval`	Eigenvalues of the hessian matrix. Used to confirm if a local minimum was indeed found. Only returned when the parameter hessian is set to TRUE.
`stderrors`	Assymptotic standard deviations of the parameters based on the observed information matrix. Only returned when the parse parameter is set to true and the hessian is indeed positive definite.

Author(s)

A. Pedro Duarte Silva

RobEstControl *Constructor function for objects of class "RobEstControl"*

Description

This function will create a control object RobEstControl containing the control parameters for the robust estimation functions fasttle, RobMxtDEst, Roblda and Robqda.

Usage

```
RobEstControl(alpha=0.75, nsamp=500,  seed=NULL, trace=FALSE, use.correction=TRUE,
  ncsteps=200, getalpha="TwoStep", getkdblstar="Twopplusone", outlin="MidPandLogR",
   trialmethod="simple", m=1, reweighted=TRUE, otpType="OnlyEst")
```

Arguments

alpha	Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*Idt@NIVar observations are used for computing the trimmed likelihood. Allowed values are between 0.5 and 1. Note that when argument 'getalpha' is set to "TwoStep" the final value of 'alpha' is estimated by a two-step procedure and the value of argument 'alpha' is only used to specify the size of the samples used in the first step.
nsamp	Number of subsets used for initial estimates.
seed	Starting value for random generator.
trace	Whether to print intermediate results.
use.correction	Whether to use finite sample correction factors.
ncsteps	The maximum number of concentration steps used each iteration of the fasttle algorithm.
getalpha	Argument specifying if the 'alpha' parameter (roughly the percentage of the sample used for computing the trimmed likelihood) should be estimadted from the data, or if the value of the argument 'alpha' should be used instead. When set to "TwoStep", 'alpha' is estimated by a two-step procedure with the value of argument 'alpha' specifying the size of the samples used in the first step. Otherwise the value of argument 'alpha' is used directly.
getkdblstar	Argument specifying the size of the initial small (in order to minimize the probability of outliers) subsets. If set to the string "Twopplusone" (default) the initial sets have twice the number of interval-value variables plus one which are they are the smaller samples that lead to a non-singular covaraince estimate). Otherwise, an integer with the size of the initial sets.
outlin	The type of outliers to be considered. "MidPandLogR" if outliers may be present in both MidPpoints and LogRanges, "MidP" if outliers are only present in MidPpoints, or "LogR" if outliers are only present in LogRanges.

`trialmethod`	The method to find a trial subset used to initialize each replication of the fasttle algorithm. The current options are "simple" (default) that simply selects 'kdblstar' observations at random, and "Poolm" that divides the original sample into 'm' non-overlaping subsets, applies the 'simple trial' and the refinement methods to each one of them, and merges the results into a trial subset.
`m`	Number of non-overlaping subsets used by the trial method when the argument of 'trialmethod' is set to 'Poolm'.
`reweighted`	Should a (Re)weighted estimate of the covariance matrix be used in the computation of the trimmed likelihood or just a "raw" covariance estimate; default is (Re)weighting.
`otpType`	The amount of output returned by fasttle. Current options are "OnlyEst" (default) where only an 'IdtE' object with the fasttle estimates is returned, "SetMD2andEst" which returns a list with an 'IdtE' object of fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and "SetMD2EstandPrfSt" wich returns a list with the previous three components plust a list of some performance statistics concerning the algorithm execution.

Value

A `RobEstControl` object

References

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

`RobEstControl`,`fasttle`, `RobMxtDEst`, `Roblda`, `Robqda`

`RobEstControl-class` *Class 'RobEstControl' - contains control parameters for the robust estimation of parametric interval data models.*

Description

This class extends the `CovControlMcd` class and contains control parameters for the robust estimation of parametric interval data models.

Objects from the Class

Objects can be created by calls of the form `new("RobEstControl", ...)` or by calling the constructor-function `RobEstControl`.

Slots

`alpha`: Inherited from class `"CovControlMcd"`. Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*Idt@NIVar observations are used for computing the trimmed likelihood. Allowed values are between 0.5 and 1.

`nsamp`: Inherited from class `"CovControlMcd"`. Number of subsets used for initial estimates. Note that when argument 'getalpha' is set to "TwoStep" the final value of 'alpha' is estimated by a two-step procedure and the value of argument 'alpha' is only used to specify the size of the samples used in the first step.

`scalefn`: Inherited from class `"CovControlMcd"` and not used in the package \sQuoteMaint.Data.

`maxcsteps`: Inherited from class `"CovControlMcd"` and not used in the package \sQuoteMaint.Data.

`seed`: Inherited from class `"CovControlMcd"`. Starting value for random generator. Default is `seed = NULL`.

`use.correction`: Inherited from class `"CovControlMcd"`. Whether to use finite sample correction factors. Default is `use.correction=TRUE`.

`trace`, `tolSolve`: Inherited from class `"CovControl"`.

`ncsteps`: The maximum number of concentration steps used each iteration of the fasttle algorithm.

`getalpha`: Argument specifying if the 'alpha' parameter (roughly the percentage of the sample used for computing the trimmed likelihood) should be estimadted from the data, or if the value of the argument 'alpha' should be used instead. When set to "TwoStep", 'alpha' is estimated by a two-step procedure with the value of argument 'alpha' specifying the size of the samples used in the first step. Otherwise, with the value of argument 'alpha' is used directly.

`getkdblstar`: Argument specifying the size of the initial small (in order to minimize the probability of outliers) subsets. If set to the string "Twopplusone" (default) the initial sets have twice the number of interval-value variables plus one (i.e., they are the smaller samples that lead to a non-singular covaraince estimate). Otherwise, an integer with the size of the initial sets.

`outlin`: The type of outliers to be consideres. "MidPandLogR" if outliers may be present in both MidPpoints and LogRanges, "MidP" if outliers are only present in MidPpoints, or "LogR" if outliers are only present in LogRanges.

`trialmethod`: The method to find a trial subset used to initialize each replication of the fasttle algorithm. The current options are "simple" (default) that simply selects 'kdblstar' observations at random, and "Poolm" that divides the original sample into 'm' non-overlaping subsets, applies the 'simple trial' and the refinement methods to each one of them, and merges the results into a trial subset.

`m`: Number of non-overlaping subsets used by the trial method when the argument of 'trialmethod' is set to 'Poolm'.

`reweighted`: Should a (Re)weighted estimate of the covariance matrix be used in the computation of the trimmed likelihood or just a "raw" covariance estimate; default is (Re)weighting.

`otpType`: The amount of output returned by fasttle. Current options are "OnlyEst" (default) where only an 'IdtE' object with the fasttle estimates is returned, "SetMD2andEst" which returns a list with an 'IdtE' object of fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and "SetMD2EstandPrfSt" wich returns a list with the previous three components plust a list of some performance statistics concerning the algorithm execution.

Extends

Class "CovControlMcd", directly. Class "CovControl" by CovControlMcd, distance 2.

Methods

No methods defined with class "RobEstControl" in the signature.

References

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

RobEstControl, fasttle, RobMxtDEst, Roblda, Robqda

RobMxtDEst-methods *Methods for Function RobMxtDEst in Package 'MAINT.Data'*

Description

Function RobMxtDEst estimates mixtures of distribution for interval-valued data using robust methods.

Usage

```
## S4 method for signature 'IData'
RobMxtDEst(Idt, grouping, Mxt=c("Hom","Het"), CovEstMet=c("Pooled","Globdev"),
    CovCase=1:4, SelCrit=c("BIC","AIC"), Robcontrol=RobEstControl(),
    l1medpar=NULL, ...)
```

Arguments

Idt	An IData object representing interval-valued entities.
grouping	Factor indicating the group to which each observation belongs to.
Mxt	Indicates the type of mixing distributions to be considered. Current alternatives are "Hom" (homocedastic) and "Het" (hetereocedasic).
CovEstMet	Method used to estimate the common covariance matrix. Alternatives are "Pooled" (default) for a pooled average of the the robust within-groups covariance estimates, and "Globdev" for a global estimate based on all deviations from the groups multivariate l1 medians. See Todorov and Filzmoser (2009) and pcaPP.l1median for details..

CovCase	Configuration of the variance-covariance matrix: a set of integers between 1 and 4.
SelCrit	The model selection criterion.
Robcontrol	A control object (S4) of class RobEstControl-class containing estimation options - same as these provided in the function specification. If the control object is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object.
l1medpar	List of named arguments to be passed to the function pcaPP.l1median used to find the multivariate l1 medians. Possible components are 'MaxStep', 'ItTol' and 'trace' (see the documentation of pcaPP.l1median for details). If kept at NULL (default) the defaults of pcaPP.l1median will be used.
...	Other named arguments.

Value

An object of type IdtMxNDRE, containing the estimation results.

References

Brito, P., Duarte Silva, A. P. (2012), Modelling Interval Data with Normal and Skew-Normal Distributions. *Journal of Applied Statistics* **39**(1), 3–20.

Hadi, A. S. and Luceno, A. (1997), Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics and Data Analysis* **25**(3), 251–272.

Todorov V. and Filzmoser P. (2009), An Object Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32**(3), 1–47.

See Also

IdtMxNDRE, RobEstControl, pcaPP.l1median.

---

stdEr--methods *Methods for function stdEr in Package 'MAINT.Data'*

Description

S4 methods for function stdEr. As in the generic stdEr S3 'miscTools' method, these methods extract standard errors of the parameter estimates, for the models fitted to Interval Data.

Usage

```
## S4 method for signature 'IdtNDE'
stdEr(x, selmodel=BestModel(x), ...)
## S4 method for signature 'IdtSNDE'
stdEr(x, selmodel=BestModel(x), ...)
## S4 method for signature 'IdtNandSNDE'
stdEr(x, selmodel=BestModel(x), ...)
```

Arguments

x	An object representing a model fitted to interval data.
selmodel	Selected model from a list of candidate models saved in object x.
...	Additional arguments for method functions.

Value

A vector of the estimated standard deviations of the parameter estimators.

See Also

vcov

---

testMod-methods *Methods for Function testMod in Package 'MAINT.Data'*

Description

Performs statistical likelihood-ratio tests that evaluate the goodness-of-fit of a nested model against a more general one.

Usage

```
testMod(ModE,RestMod=ModE@ModelConfig[2]:length(ModE@ModelConfig),FullMod="Next")
```

Arguments

ModE	An object of class IdtE representing the estimates of a model fitted to a data set of interval-value variables
RestMod	Indices of the restricted models beeing evaluated in the NULL hypothesis
FullMod	Either indices of the general models beeing evaluated in the alternative hypothesis or the strings "Next" (default) or "All". In the former case a Restricted model is always compared against the most parsimonious alternative that encompasses it, and in latter all possible comparisons are performed

Value

An object of type ConfTests with the results of the tests performed

Examples

```
# Create an Interval-Data object containing the intervals of temperatures by quarter
# for 899 Chinese meteorological stations.

ChinaT <- IData(ChinaTemp[1:8])

# Estimate by maximum likelihood the parameters of Gaussian models
# for the Winter (1st and 4th) quarter intervals

ChinaWTE <- mle(ChinaT[,c(1,4)])
cat("China maximum likelhiood estimation results for Winter quarters:\n")
print(ChinaWTE)

# Perform Likelihood-Ratio tests comparing models with consecutive nested Configuration
testMod(ChinaWTE)

# Perform Likelihood-Ratio tests comparing all possible models
testMod(ChinaWTE,FullMod="All")

# Compare model with covariance Configuration case 3 (MidPoints independent of LogRanges)
# against model with covariance Configuration 1 (unrestricted covariance)
testMod(ChinaWTE,RestMod=3,FullMod=1)
```

---

vcov--methods *Methods for function vcov in Package 'MAINT.Data'*

Description

S4 methods for function vcov. As in the generic vcov S3 'stats' method, these methods extract variance-covariance estimates of parameter estimators, for the models fitted to Interval Data.

Usage

```
## S4 method for signature 'IdtNDE'
vcov(object, selmodel=BestModel(object), ...)
## S4 method for signature 'IdtSNDE'
vcov(object, selmodel=BestModel(object), ...)
## S4 method for signature 'IdtNandSNDE'
vcov(object, selmodel=BestModel(object), ...)
## S4 method for signature 'IdtMxNDE'
vcov(object, selmodel=BestModel(object), group=NULL, ...)
## S4 method for signature 'IdtMxSNDE'
vcov(object, selmodel=BestModel(object), group=NULL, ...)
```

Arguments

object	An object representing a model fitted to interval data.
selmodel	Selected model from a list of candidate models saved in object.
group	Ther group for each the estimated parameter variance-covariance will be returned. If NULL (default), "vcov" will return a three-dimensional array with a matrix of the estimated covariances between the parameter estimates for each group at each level of the third dimension.
...	Additional arguments for method functions.

Value

For the "IdtNDE", "IdtSNDE" and "IdtSNDE" methods: a matrix of the estimated covariances between the parameter estimates. For the "IdtMxNDE", and "IdtMxSNDE" methods: if argument "group" is set to NULL, a three-dimensional array with a matrix of the estimated covariances between the parameter estimates for each group at each level of the third dimension. If argument "group" is set to an integer, the matrix with the estimated covariances between the parameter estimates, for the group chosen.

See Also

stdEr

Index