Package ‘MAINT.Data’
August 11, 2018

Type       Package
Title      Model and Analyse Interval Data
Version    1.2.3
Date       2018-08-10
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, mclust
LinkingTo  Rcpp, RcppEigen
NeedsCompilation yes
Repository  CRAN
Date/Publication 2018-08-10 23:00:02 UTC

R topics documented:

MAINT.Data-package ........................................ 3
BestModel-methods ....................................... 7
Cars ......................................................... 8
ChinaTemp .................................................. 8
coei–methods ............................................. 9
R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfTests-class</td>
<td>10</td>
</tr>
<tr>
<td>cor-methods</td>
<td>11</td>
</tr>
<tr>
<td>DACrossVal</td>
<td>12</td>
</tr>
<tr>
<td>EMControl</td>
<td>14</td>
</tr>
<tr>
<td>EMControl-class</td>
<td>14</td>
</tr>
<tr>
<td>extmatrix-class</td>
<td>15</td>
</tr>
<tr>
<td>fasttle-methods</td>
<td>15</td>
</tr>
<tr>
<td>fulltle-methods</td>
<td>19</td>
</tr>
<tr>
<td>getIdOutl</td>
<td>22</td>
</tr>
<tr>
<td>IData</td>
<td>23</td>
</tr>
<tr>
<td>IData-class</td>
<td>24</td>
</tr>
<tr>
<td>IdtE-class</td>
<td>26</td>
</tr>
<tr>
<td>Idtlda-class</td>
<td>27</td>
</tr>
<tr>
<td>IdtMANOVA-class</td>
<td>28</td>
</tr>
<tr>
<td>IdtMclust-class</td>
<td>30</td>
</tr>
<tr>
<td>Idtmclust-methods</td>
<td>31</td>
</tr>
<tr>
<td>IdtMxE-class</td>
<td>33</td>
</tr>
<tr>
<td>IdtMxNandSNDE-class</td>
<td>34</td>
</tr>
<tr>
<td>IdtMxNDE-class</td>
<td>35</td>
</tr>
<tr>
<td>IdtMxNDRE-class</td>
<td>37</td>
</tr>
<tr>
<td>IdtMxSNDE-class</td>
<td>38</td>
</tr>
<tr>
<td>IdtMxNDE-class</td>
<td>40</td>
</tr>
<tr>
<td>IdtNandSNDE-class</td>
<td>40</td>
</tr>
<tr>
<td>IdtNDE-class</td>
<td>41</td>
</tr>
<tr>
<td>Idtqda-class</td>
<td>41</td>
</tr>
<tr>
<td>IdtSNDE-class</td>
<td>42</td>
</tr>
<tr>
<td>IdtSNgenda-class</td>
<td>43</td>
</tr>
<tr>
<td>IdtSngNandSNDE-class</td>
<td>44</td>
</tr>
<tr>
<td>IdtSngNDE-class</td>
<td>45</td>
</tr>
<tr>
<td>IdtSngNDRE-class</td>
<td>47</td>
</tr>
<tr>
<td>IdtSngSNDE-class</td>
<td>48</td>
</tr>
<tr>
<td>IdtSNlocda-class</td>
<td>50</td>
</tr>
<tr>
<td>lda-methods</td>
<td>51</td>
</tr>
<tr>
<td>LoansbyPurpose_minmaxDt</td>
<td>53</td>
</tr>
<tr>
<td>LoansbyRiskLvs_minmaxDt</td>
<td>54</td>
</tr>
<tr>
<td>LoansbyRiskLvs_qntlDt</td>
<td>55</td>
</tr>
<tr>
<td>LRTest-class</td>
<td>56</td>
</tr>
<tr>
<td>MANOVA-methods</td>
<td>56</td>
</tr>
<tr>
<td>mle-methods</td>
<td>58</td>
</tr>
<tr>
<td>qda-methods</td>
<td>60</td>
</tr>
<tr>
<td>qHardRoqF</td>
<td>61</td>
</tr>
<tr>
<td>RepLOptim</td>
<td>62</td>
</tr>
<tr>
<td>Robda-methods</td>
<td>65</td>
</tr>
<tr>
<td>RobEstControl</td>
<td>67</td>
</tr>
<tr>
<td>RobEstControl-class</td>
<td>70</td>
</tr>
<tr>
<td>RobMxtDEst-methods</td>
<td>72</td>
</tr>
<tr>
<td>snda-methods</td>
<td>73</td>
</tr>
<tr>
<td>stdEr-methods</td>
<td>76</td>
</tr>
</tbody>
</table>
**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, Discriminant Analysis and Gaussian Model Based Clustering.

**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 (Duarte Silva, Filzmoser and Brito (2017)) estimation, and statistical tests for the different considered configurations. Methods for (M)ANOVA, Discriminant Analysis (Duarte Silva and Brito (2015)) and model based clustering (Brito, Duarte Silva and Dias (2015)) of this data class are also provided.
Author(s)
Pedro Duarte Silva <psilva@porto.ucp.pt>
Paula Brito <mpbrito.fep.up.pt>
Maintainer: Pedro Duarte Silva <psilva@porto.ucp.pt>

References


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 distribution 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 likelihood estimation results:\n")
print(ChinaWTE.C1)
cat("Standard Errors of Estimators:\n") ; print(stdEr(ChinaWTE.C1))

# Estimate normal distribution 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 likelihood estimation results:\n")
print(ChinaWTE.C234)
cat("Standard Errors of Estimators:\n") ; print(stdEr(ChinaWTE.C234))

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

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

# Estimate skew-normal distribution parameters

ChinaWTE.SkN <- mle(ChinaWT,Model="SKNormal")
cat("Winter temperatures of China -- Skew-Normal maximum likelihood 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

CV1lda <- DACrossVal(ChinaWT,ChinaTemp$GeoReg,TrainAlg=lda, CovCase=BestModel(HiRes(ManvChinaWT.C13)),CVrep=1)
suonym(CV1lda[,"C1err"])
glberrors <-
apply(CVlda[,"Nk"]*CVlda[,"Clerr"],1,sum)/apply(CVlda[,"Nk"],1,sum)
cat("Average global classification error =",mean(glerrors),"\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.

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

# Display the first and last observations

head(CarsIdt)
tail(CarsIdt)

# Estimate normal distribution parameters

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

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

## Not run:

CarsTE <- fulltlte(CarsIdt)
cat("Cars data -- normal maximum trimmed likelihood estimation results:\n")
print(CarsTE)

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

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

## End(Not run)

# Create an Interval-Data object containing the intervals of loan data # (from the Kaggle Data Science platform) aggregated by loan purpose

LbyPIdt <- IData(LoansbyPurpose_minmaxDt,
                 VarNames=c("ln-inc","ln-revolbal","open-acc","total-acc"))

print(LbyPIdt)
## 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. “IdtCrt” stands for the criterion originally used in the ModE estimation, while “BIC” and “AIC” represent respectively the Bayesian and Akaike information criteria.

## Value

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

**Cars Data Set**

**Description**

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

**Usage**

data(Cars)

**Format**

A data frame containing 27 observations on 9 variables, the first eight with the lower and upper bounds 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 meteorological 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 geographical region of each station.
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. Only used when object has class `IdtSNDE` or `IdtNandSNDE` and in this latter case when argument “selmodel” chooses a Skew-Normal model. 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


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

\textbf{TestRes:} List of test results; each element is an object of class 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.

\textbf{RestModels:} The restricted model (corresponding to the null hypothesis)

\textbf{FullModels:} The full model (corresponding to the alternative hypothesis)

Methods

\texttt{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

\texttt{mle,IData,LRTest}
Methods for function cor in Package 'MAINT.Data'

Description

S4 methods for function cor. These methods extract estimates of correlation matrices for the models fitted to Interval Data.

Usage

```r
## S4 method for signature 'IdtNDE'
cor(x)
## S4 method for signature 'IdtSNDE'
cor(x)
## S4 method for signature 'IdtNandSNDE'
cor(x)
## S4 method for signature 'IdtMxNDE'
cor(x)
## S4 method for signature 'IdtMxSNDE'
cor(x)
```

Arguments

- `x` An object representing a model fitted to interval data.

Value

For the `IdtNDE`, `IdtSNDE` and `IdtNandSNDE` methods or `IdtMxNDE`, `IdtMxSNDE` methods with slot “Hmcdt” equal to TRUE: a matrix with the estimated correlations.

For the `IdtMxNDE`, and `IdtMxSNDE` methods with slot “Hmcdt” equal to FALSE: a three-dimensional array with a matrix with the estimated correlations for each group at each level of the third dimension.

See Also

- `var`
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

```r
# 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)
```
Constructor function for objects of class EMControl

Description

This function will create a control object of class EMControl containing the control parameters for the EM algorithm used in estimation of Gaussian mixtures by function idtmclust.

Usage

EMControl(nrep=100, maxiter=1000, convtol=0.01, protol=1e-6, seed=NULL)

Arguments

- **nrep**: Number of replications (different randomly generated starting points) of the EM algorithm.
- **maxiter**: Maximum number of iterations in each replication of the EM algorithm.
- **convtol**: Numeric tolerance for testing the convergence of the EM algorithm. Convergence is assumed when the log-likelihood changes less than convtol.
- **protol**: Numeric tolerance for the mixture proportions. Proportions below protol, considered to be zero, are not allowed.
- **seed**: Starting value for random generator.

Value

An EMControl object

See Also

Idtmclust

EM algorithm control parameters for fitting Gaussian mixtures to interval data.

Description

This class contains the control parameters for the EM algorithm used in estimation of Gaussian mixtures by function Idtmclust.

Objects from the Class

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

nrep  Number of replications (different randomly generated starting points) of the EM algorithm.
maxiter  Maximum number of iterations in each replication of the EM algorithm.
convtol  Numeric tolerance for testing the convergence of the EM algorithm. Convergence is assumed when the log-likelihood changes less than convtol.
protol  Numeric tolerance for the mixture proportions. Proportions below protol, considered to be zero, are not allowed.
seed  Starting value for random generator.

See Also

EMControl

extmatrix-class  Class “extmatrix”

Description

“extmatrix” is a simple extension of the base matrix class, 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,
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 * nrow(Idt) observations are used for computing the trimmed likelihood. 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. Allowed values are between 0.5 and 1.

nsamp
Number of subsets used for initial estimates.

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.

rawMD2Dist
The assumed reference distribution of the raw MCD squared distances, which is used to find to cutoffs defining the observations kept in one-step reweighted MCD estimates. Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual Qui-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005).

MD2Dist
The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual Qui-squared, or the Beta and F distributions proposed by Cerioli (2010).
eta
Nominal size for the null hypothesis that a given observation is not an outlier. Defines the raw MCD Mahalanobis distances cutoff used to choose the observations kept in the reweightening step.

multiCmpCor
Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at ‘eta’ nominal level. ‘always’ – testing all n entities at 1.- (1-’eta’^(1/n)); and ‘iterstep’ – use the iterated rule proposed by Cerioli (2010), i.e., make an initial set of tests using the nominal size 1.- (1-’eta’^(1/n)), and if no outliers are detected stop. Otherwise, make a second step testing for outliers at the ‘eta’ nominal level.

getakdblstar
Argument specifying the size of the initial small (in order to minimize the probability of outliers) subsets. If set to the string “Twoopplusone” (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 MidP-points, 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-overlapping 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-overlapping 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 “SetMD2andEst” (default) which returns an ‘IdtSngNDRE’ object with the fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and “SetMD2EstandPrfSt” which returns an ‘IdtSngNDRE’ object with the previous slots plus 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.

Value
An object of class IdtE with the fasttle estimates, the value of the comparison criterion used to select the covariance configurations, the robust squared Mahalanobis distances, and optionally (if argument ‘otpType’ is set to true) performance statistics concerning the algorithm execution.
References


See Also

`fulltle, RobEstControl, getIdtOutl, IdtSngNDRE`

Examples

```r
## 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 by the fast trimmed maximum likelihood estimator,
# using a two-step procedure to select the trimming parameter, a reweighted
# MCD estimate, and the classical 97.5% chi-squared quantile cut-offs.
Chinafasttle1 <- fasttle(ChinaT)
cat("China maximum trimmed likelihood estimation results =\n")
print(Chinafasttle1)

# Estimate parameters by the fast trimmed maximum likelihood estimator, using
# the trimming parameter that maximizes breakdown, and a reweighted MCD estimate
# based on the 97.5% quantiles of Hardin and Rocke adjusted F distributions.
Chinafasttle2 <- fasttle(ChinaT, alpha=0.5, getalpha=FALSE, rawMDDist="HardRockeAdjF")
cat("China maximum trimmed likelihood estimation results =\n")
print(Chinafasttle2)
```
fulltle-methods

# Estimate parameters by the fast trimmed maximum likelihood estimator, using a two-step procedure
# to select the trimming parameter, a reweighted MCD estimate based on Hardin and Rocke adjusted
# F distributions, and 95% quantiles, and the Cerioli Beta and F distributions together
# with Cerioli iterated procedure to identify outliers in the first step.

Chinafasttle3 <- fasttle(ChinaT, rawMD2Dist="HardRockeAdjF", eta=0.05, MD2Dist="CerioliBetaF",
multiCmpCor="iterstep")
cat("China maximum trimmed likelihood estimation results =\n")
print(Chinafasttle3)

## End(Not run)

---

fulltle-methods

Methods for Function fulltle in Package 'MAINT.Data'

Description

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

Usage

fulltle(Idt, CovCase=1:4, SelCrit=c("BIC","AIC"), alpha=0.75,
use.correction=TRUE, getalpha="TwoStep",
rawMD2Dist=c("ChiSq","HardRockeAsF","HardRockeAdjF"),
MD2Dist=c("ChiSq","CerioliBetaF"),
etta=0.025, multiCmpCor=c("never","always","iterstep"),
outlin=c("MidPandLogR","MidP","LogR"), reweighted=TRUE,
force=FALSE, ...)

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*nrow(Idt) observations are used for computing the trimmed likelihood. 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. Allowed values are between 0.5 and 1.
use.correction whether to use finite sample correction factors; defaults to TRUE.
**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.

**rawMD2Dist**  The assumed reference distribution of the raw MCD squared distances, which is used to find to cutoffs defining the observations kept in one-step reweighted MCD estimates. Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual Qui-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005).

**MD2Dist**  The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual Qui-squared, and the Beta and F distributions proposed by Cerioli (2010).

**eta**  Nominal size of the null hypothesis that a given observation is not an outlier. Defines the raw MCD Mahalanobis distances cutoff used to choose the observations kept in the reweighting step.

**multiCmpCor**  Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at the ‘eta’ nominal level. ‘always’ – testing all n entities at 1.- (1.-’eta’^((1/n))); and ‘iterstep’ – use the iterated rule proposed by Cerioli (2010), i.e., make an initial set of tests using the nominal size 1.- (1.-’eta’^((1/n))), and if no outliers are detected stop. Otherwise, make a second step testing for outliers at the ‘eta’ nominal level.

**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 MidP-points, or “LogR” if outliers are only present in LogRanges.

**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.

**force**  A boolean flag indicating whether, for moderate or large data sets the algorithm should proceed anyway, regardless of an expected long execution time, due to exponential explosions in the number of different subsets that need to be evaluated by fulltle

...  Further arguments to be passed to internal functions of ‘fulltle’.

**Value**

An object of class **IdtE** with the fasttle estimates, the value of the comparison criterion used to select the covariance configurations and the robust squared Mahalanobis distances.

**References**


### See Also

`fasttle`, `getIdOutl`

### Examples

```r
## Not run:

# Create an Interval-Data object containing the intervals for characteristics
# of 27 cars models.
CarsIdt <- IData(Cars[1:8], VarNames=c("Price","EngineCapacity","TopSpeed","Acceleration"))

# Display the first and last observations
print(head(CarsIdt))
print(tail(CarsIdt))

# Estimate parameters by the full trimmed maximum likelihood estimator,
# using a two-step procedure to select the trimming parameter, a reweighted
# MCD estimate, and the classical 97.5% chi-squared quantile cut-offs.
CarsTE1 <- fulltle(CarsIdt)
cat("Cars data -- normal maximum trimmed likelihood estimation results:
print(CarsTE1)

# Estimate parameters by the full trimmed maximum likelihood estimator, using
# the trimming parameter that maximizes breakdown, and a reweighted MCD estimate
# based on the 97.5% quantiles of Hardin and Rocke adjusted F distributions.
CarsTE2 <- fulltle(CarsIdt, alpha=0.5, getalpha=FALSE, rawMD2Dist="HardRockeAdjF")
cat("Cars data -- normal maximum trimmed likelihood estimation results:
print(CarsTE2)

# Estimate parameters by the full trimmed maximum likelihood estimator, using
```
# a two-step procedure to select the trimming parameter, and a reweighted MCD estimate
# based on Hardin and Rocke adjusted F distributions, 95% quantiles, and
# the Cerioli Beta and F distributions together with his iterated procedure
# to identify outliers in the first step.

CarsTE3 <- fulltltte(CarsIdt, rawMD2Dist="HardRockeAdjF", eta=0.05, MD2Dist="CerioliBetaF",
  multiCmpCor="iterstep")
cat("Cars data -- normal maximum trimmed likelihood estimation results:\n")
print(CarsTE3)

## End(Not run)

---

### getIdOutl

Get Interval Data Outliers

#### Description

Identifies outliers in a data set of Interval-valued variables

#### Usage

```r
getIdOutl(Idt, IdtE=NULL, muE=NULL, SigE=NULL, eta=0.025, Rewind=NULL, m=length(Rewind),
  RefDist=c("ChiSq","HardRockeAdjF","HardRockeAsF","CerioliBetaF"),
  multiCmpCor=c("never","always","iterstep"),
  outlin=c("MidPandLogR","MidP","LogR"))
```

#### Arguments

- **Idt**: An IData object representing interval-valued entities.
- **IdtE**: An object of class IdtSngNDRE or IdtSngNDE containing mean and covariance estimates.
- **muE**: Vector with the mean estimates used to find Mahalanobis distances. When specified, it overrides the mean estimate supplied in “IdtE”.
- **SigE**: Matrix with the covariance estimates used to find Mahalanobis distances. When specified, it overrides the covariance estimate supplied in “IdtE”.
- **eta**: Nominal size of the null hypothesis that a given observation is not an outlier.
- **Rewind**: A vector with the subset of entities used to compute trimmed mean and covariance estimates when using a reweighted MCD. Only used when the ‘RefDist’ argument is set to “CerioliBetaF”
- **m**: Number of entities used to compute trimmed mean and covariance estimates when using a reweighted MCD. Not used when the ‘RefDist’ argument is set to “ChiSq.”
multicmpcor  Whether a multicomparison correction of the nominal size (eta) for the outliers
tests should be performed. Alternatives are: ‘never’ – ignoring the multicom-
parisons and testing all entities at the ‘eta’ nominal level. ‘always’ – testing all
entities at 1.- (1.-‘eta’^n(1/n)); and ‘iterstep’ – use the iterated rule proposed
by Cerioli (2010), i.e., make an initial set of tests using the nominal size 1.- (1-
‘eta’^n(1/n)), and if no outliers are detected stop. Otherwise, make a second step
testing for outliers at the ‘eta’ nominal level.
RefDist  The assumed reference distributions used to find cutoffs defining the observa-
tions assumed as outliers. Alternatives are “ChiSq”, “HardRockeAsF”, “HardRocke-
AdjF” and “CerioliBetaF”, respectively for the usual Qui-squared, the asymp-
totic and adjusted scaled F distributions proposed by Hardin and Rocke (2005),
and the Beta and F distributions proposed by Cerioli (2010).
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 MidP-
points, or “LogR” if outliers are only present in LogRanges.

Value
A vector with the indices of the entities identified as outliers.

References
Cerioli, A. (2010), Multivariate Outlier Detection with High-Breakdown Estimators. Journal of the
American Statistical Association 105 (489), 147–156.
in Data Analysis and Classification, 1–38.
Hardin, J. and Rocke, A. (2005), The Distribution of Robust Distances. Journal of Computational
and Graphical Statistics 14, 910–927.

See Also
fasttle, fulltle

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 class 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\n")
print(tail(ChinaT,n=3))
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

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 Interval Variables.
rownames signature(x = "IData"): returns the row (entity) names for an object of class IData.
colnames signature(x = "IData"): returns column (variable) names for an object of class IData.
head signature(x = "IData"): head 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.
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


See Also

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

IdtE-class

Class IdtE

Description

IdtE contains estimation results for the models assumed for single distributions, or mixtures of distributions, underlying data sets of interval-valued entities.

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

logLik: 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**


**See Also**

- mle, fasttle, fulltle, MANOVA, RobMxtDEst, IDdata

---

**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 IDtlda-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


See Also

qda, MANOVA, Roblda, Robqda, snda, I0data

---

**IdtMANOVA-class**

**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 particular, the following specializations of IdtMANOVA are currently implemented:

- **IdtClMANOVA** extends IdtMANOVA, assuming a classical (i.e., homocedastic gaussian) setup.
- **IdtHetMANOVA** extends IdtMANOVA, assuming a heterocedastic gaussian set-up.
- **IdtLocSNMANOVA** 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.
- **IdtGenSNMANOVA** extends IdtMANOVA, assuming a Skew-Normal general model set-up.
- **IdtGenNSNMANOVA** extends IdtMANOVA, assuming either a heterocedastic gaussian or Skew-Normal general model set-up.
IdtMANOVA-class

Slots

IVar: 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.
QuiSqr: 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 = "IdtLocNSMANOVA"): Linear Discriminant Analysis using the estimated model parameters.
qda signature(x = "IdtHetNSMANOVA"): Quadratic Discriminant Analysis using the estimated model parameters.
qda signature(x = "IdtGenNSMANOVA"): Quadratic Discriminant Analysis using the estimated model parameters.
snda signature(x = "IdtLocNSMANOVA"): 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 = "IdtGenNSMANOVA"): Discriminant Analysis using maximum likelihood parameter estimates of SkewNormal mixtures assuming a general model (i.e., groups differ in all parameters).
snda signature(x = "IdtGenNSMANOVA"): 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


See Also

MANOVA, lda, qda, snda, IData

idtMclust-class  

Class IdtMclust

Description

IdtMclust contains the results of fitting mixtures of Gaussian distributions to interval data represented by objects of class IData.

Slots

call: The matched call that created the IdtMclust object
data: The IData data object
Nobs: Number of entities under analysis (cases)
NIVar: Number of interval variables
SelCrit: The model selection criterion; currently, AIC and BIC are implemented
Homcdt: Indicates whether the optimal model corresponds to a homocedastic (TRUE) or a heterocedastic (FALSE) setup
BestG: The optimal number of mixture components.
BestC: The configuration case of the variance-covariance matrix in the optimal model
logLiks: The logarithms of the likelihood function for the different models tried
logLik: The logarithm of the likelihood function for the optimal model
AICs: The values of the AIC criterion for the different models tried
aic: The value of the AIC criterion for the he optimal model
BICs: The values of the BIC criterion for the different models tried
bic: The value of the BIC criterion for the he optimal model
parameters A list with the following components:
pro A vector whose \( k \)th component is the mixing proportion for the \( k \)th component of the mixture model.
**mean**  The mean for each component. If there is more than one component, this is a matrix whose kth column is the mean of the kth component of the mixture model.

**covariance**  A three-dimensional array with the covariance estimates. If Hmcdt is FALSE (heterocedastic setups) the third dimension levels run through the BestG mixture components, with one different covariance matrix for each level. Otherwise (homocedastic setups), there is only one covariance matrix and the size of the third dimension equals one.

z:  A matrix whose [i,k]th entry is the probability that observation i in the test data belongs to the kth class.

classification:  The classification corresponding to z, i.e. map(z).

allres:  A list with the detailed results for all models fitted.

**Methods**

**show**  signature(object = "IdtMclust"): show S4 method for the IdtMclust-class

**summary**  signature(object = "IdtMclust"): summary S4 method for the IdtMclust-class

**Author(s)**

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

**References**


**See Also**

*Idtmclust*

---

**Description**

Performs Gaussian model based clustering for interval data

**Usage**

Idtmclust(Idt, G = 1:9, CovCase=1:4, SelCrit=c("BIC","AIC"), Mxt=c("Hom","Het","HomandHet"), control=EMControl())
Arguments

**Idt**
An IData object representing interval-valued entities.

**G**
An integer vector specifying the numbers of mixture components (clusters) for which the BIC is to be calculated.

**CovCase**
Configuration of the variance-covariance matrix: a set of integers between 1 and 4.

**SelCrit**
The model selection criterion.

**control**
A list of control parameters for EM. The defaults are set by the call `EMControl()`.

**Mxt**
The type of Gaussian mixture assumed by Idtmclust. Alternatives are “Hom” (default) for homocedastic mixtures, “Het” for heterocedastic mixtures, and “HomandHet” for both homocedastic and heterocedastic mixtures.

Value

An object of class `IdtMclust` providing the optimal (according to BIC) mixture model estimation.

References


See Also

`IdtMclust`, `EMControl`, `EMControl`

Examples

```r
# Create an Interval-Data object containing the intervals of loan data
# (from the Kaggle Data Science platform) aggregated by loan purpose

LbyPIdt <- IData(LoansbyPurpose_minmaxDt,
  VarNames=c("ln-inc","ln-revolbal","open-acc","total-acc"))

print(LbyPIdt)

## Not run:

# Fit homocedastic Gaussian mixtures with up to four components

mclustres <- IdtMclust(LbyPIdt,G=1:4)
```
IdtMxE-class

print(mclustres)

# Display the results of the best mixture according to the BIC
summary(mclustres, parameters=TRUE, classification=TRUE)

## End(Not run)

---

### IdtMxE-class

**Class IdtMxE**

---

### Description

IdtMxE extends the IdtE class, assuming that the data can be characterized by a mixture of distributions, for instances considering partitions of entities into different groups.

### 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 distributions 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


See Also

*IdtE, IdtSngDE, 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
- **aic**: Inherited from class *IdtE*. Value of the AIC criterion
- **bic**: Inherited from class *IdtE*. Value of the BIC criterion
**IdtMxNDE-class**

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 distributions 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**


**See Also**

`IdtE`, `IdtMxE`, `IdtSngNandSNDE`, `MANOVA`, `RobMxTDEst`, `IData`
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

logLik: 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 `IdtE`, by class `IdtMxE`, distance 2.

**Methods**

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

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

**Author(s)**

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

**References**


**See Also**

`IdtE, IdtMxE, IdtMxNDE, IdtSngNDE, IData, MANOVA`
IdtMxNDRE-class

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)
RobNmve: 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
rawSet A vector with the trimmed subset elements used to compute the raw (not reweighted) MCD covariance estimate for the chosen configuration.
RewghtdSet 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.
cnpR A vector of length two containing the consistency correction factor and the finite sample correction factor of the final estimate of the covariance matrix.
raw.cov A matrix with the raw MCD estimator used to compute the robust squared Mahalanobis distances of RobMD2.
IdtMxSNDE-class

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
- `logLik`: 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 `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


See Also

IdtE, IdtMxE, IdtSngSNDE, MANOVA, IData

IdtMxtNDE-class

Description

IdtMxtNDE is an union of classes IdtMxNDE and IdtMxNDRE, containing the results of mixture Normal model parameter estimation by maximum likelihood (IdtMxNDE) or robust (IdtMxNDRE) methods.

See Also

IdtE, IdtMxE, IdtMxNDE, IdtMxNDRE

IdtNandSNDE-class

Description

IdtNandSNDE is a union of classes IdtsngNandSNDE and IdtMxnandsnde, 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 class IdtNandSNDE

stdEr signature(x = "IdtNandSNDE"): extracts standard errors from objects of class IdtNandSNDE

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

mean signature(x = "IdtNandSNDE"): extracts the mean vector estimate from objects of class IdtNandSNDE

var signature(x = "IdtNandSNDE"): extracts the variance-covariance matrix estimate from objects of class IdtNandSNDE

cor signature(x = "IdtNandSNDE"): extracts the correlation matrix estimate from objects of class IdtNandSNDE

References


See Also

IData, mle, fasttle, fulltle, MANOVA, RobMxtDEst, IdtSngNandSNDE, IdtMxnandsnde
IdtNDE-class

**Description**

IdtNDE is a union of classes `IdtSngNDE`, `IdtSngNDRE`, `IdtMxNDE` and `IdtMxNDRE`, used for storing the estimation results of Normal modelizations for Interval Data.

**Methods**

- `coef` signature(coef = "IdtNDE"): extracts parameter estimates from objects of class IdtNDE
- `stdEr` signature(x = "IdtNDE"): extracts standard errors from objects of class IdtNDE
- `vcov` signature(x = "IdtNDE"): extracts an estimate of the variance-covariance matrix of the parameters estimators for objects of class IdtNDE
- `mean` signature(x = "IdtNDE"): extracts the mean vector estimate from objects of class IdtNDE
- `var` signature(x = "IdtNDE"): extracts the variance-covariance matrix estimate from objects of class IdtNDE
- `cor` signature(x = "IdtNDE"): extracts the correlation matrix estimate from objects of class IdtNDE

**References**


**See Also**

`IdtSngNDE`, `IdtSngNDRE`, `IdtMxNDE`, `IdtMxNDRE`, `IdtSNDE`, `IData`, `mle`, `fasttlice`, `fulltlice`, `MANOVA`, `RobMxtDEst`

Idtqda-class

**Description**

Idtqda contains the results of Quadratic Discriminant Analysis for the interval data
IdtSNDE-class

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, 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


See Also

qda, MANOVA, Robqda, IData

---

IdtSNDE-class

Class "IdtSNDE"

Description

IdtSNDE is a class union of classes IdtSngSNDE and IdtMxSNDE, used for storing the estimation results of Skew-Normal modelizations for Interval Data.
Methods

coeff signature(coeff = "IdtSNDE"): extracts parameter estimates from objects of class IdtSNDE

stdErr signature(x = "IdtSNDE"): extracts standard errors from objects of class IdtSNDE

vcov signature(x = "IdtSNDE"): extracts an asymptotic estimate of the variance-covariance matrix of the parameter estimators for objects of class IdtSNDE

mean signature(x = "IdtSNDE"): extracts the mean vector estimate from objects of class IdtSNDE

var signature(x = "IdtSNDE"): extracts the variance-covariance matrix estimate from objects of class IdtSNDE

cor signature(x = "IdtSNDE"): extracts the correlation matrix estimate from objects of class IdtSNDE

References


See Also

IdtSNgenda, mle, MANOVA, IdtSNgSnde, IdtMxSNDE, IdtNDE

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 skewness 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 skewness parameter ("gamma1") estimates for each group.

det: 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


See Also

`MANOVA`, `snda`, `IData`
**IdtSngNDE-class**

- **NIVar**: Inherited from class `IdtE`. Number of interval variables
- **SelCrit**: Inherited from class `IdtE`. The model selection criterion; currently, AIC and BIC are implemented
- **logLik**: 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**


**See Also**

`idata`, `IdtMxNandSNDE`, `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: Vector with the maximum likelihood mean vectors estimates
mleNmuEse: Vector with the maximum likelihood means’ standard errors
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


See Also

IData, mle, IdtSngNDRE, IdtSngSNDE, IdtMxNDE
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
logLik: 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
rawSet A vector with the trimmed subset elements used to compute the raw (not reweighted) MCD covariance estimate for the chosen configuration.
RewghtdSet A vector with the final trimmed subset elements used to compute the fasttłe estimates.
RobMD2 A vector with the robust squared Mahalanobis distances used to select the trimmed subset.
cnp2 A vector of length two containing the consistency correction factor and the finite sample correction factor of the final estimate of the covariance matrix.
raw.cov A matrix with the raw MCD estimator used to compute the robust squared Mahalanobis distances of RobMD2.
raw.cnp2 A vector of length two containing the consistency correction factor and the finite sample correction factor of the raw estimate of the covariance matrix.
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 by fasttle algorithm by replication.
- **StpLogLik**: A list with one component by Covariance Configuration, containing a matrix with the evolution of the log-likelihoods found by fasttle algorithm by replication and refinement step.

**Extends**


**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**


**See Also**

*IData, fasttle, fulltle, IdtSngNDE, IdtMxNDRE*

---

**IdtSngSNDE-class**

**Class IdtSngSNDE**

**Description**

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

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
- **logLik**: 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


See Also

- mle, IData, IdtSngNDE, IdtMxSNDE
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


See Also

snda, MANOVA, IData

Description

Lda performs linear discriminant analysis of Interval Data based on classic estimates of a mixture of Gaussian models.

Usage

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

Arguments

- **x**: An object of class `IData`, `IdtMxtNDE`, `IdtClMANOVA` or `IdtLocNSNMANOVA` with either the original Interval Data, an estimate of a mixture of gaussian models for Interval Data, or the results of an Interval Data 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.
- **egvtol**: Tolerance level for the eigenvalues of the product of the inverse within by the between covariance matrices. When an eigenvalue has an absolute value below egvtol, it is considered to be zero.
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 whether a warning message should be printed if the method fails.
se樊model      Selected model from a list of candidate models saved in object x.
...        Other named arguments.

References


See Also

qda, snida, Roblda, Robqda, IData, IdtMxtNDE, IdtClMANOVA, IdtLocNSMANOVA

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"])
errors <-
apply(CVlda[,"Nk"]*CVlda[,"Clerr"],1,sum)/apply(CVlda[,"Nk"],1,sum)
cat("Average global classification error =",mean(errors),"\n")

## End(Not run)
Loans by purpose: minimum and maximum Data Set

Description

This data set consist of the lower and upper bounds of the intervals for four interval characteristics of the loans aggregated by their purpose. The original microdata is available at the Kaggle Data Science platform and consists of 887 383 loan records characterized by 75 descriptors. Among the large set of variables available, we focus on borrowers’ income and account and loan information aggregated by the 14 loan purposes, which are considered as the units of interest.

Usage

data(LoansbyPurpose_minmaxDt)

Format

A data frame containing 14 observations on the following 8 variables.

- **ln-inc_min** The minimum, for the current loan purpose, of natural logarithm of the self-reported annual income provided by the borrower during registration.
- **ln-inc_max** The maximum, for the current loan purpose, of natural logarithm of the self-reported annual income provided by the borrower during registration.
- **ln-revolbal_min** The minimum, for the current loan purpose, of natural logarithm of the total credit revolving balance.
- **ln-revolbal_max** The maximum, for the current loan purpose, of natural logarithm of the total credit revolving balance.
- **open-acc_min** The minimum, for the current loan purpose, of the number of open credit lines in the borrower’s credit file.
- **open-acc_max** The maximum, for the current loan purpose, of the number of open credit lines in the borrower’s credit file.
- **total-acc_min** The minimum, for the current loan purpose, of the total number of credit lines currently in the borrower’s credit file.
- **total-acc_max** The maximum, for the current loan purpose, of the total number of credit lines currently in the borrower’s credit file.

Source

https://www.kaggle.com/wendykan/lending-club-loan-data
Loans by risk levels: minimum and maximum Data Set

Description

This data set consist of the lower and upper bounds of the intervals for four interval characteristics for 35 risk levels (from A1 to G5) of loans. The original microdata is available at the Kaggle Data Science platform and consists of 887 383 loan records characterized by 75 descriptors. Among the large set of variables available, we focus on borrowers’ income and account and loan information aggregated by the 35 risk levels which are considered as the units of interest.

Usage

data(LoansbyRiskLvs_minmaxDt)

Format

A data frame containing 35 observations on the following 8 variables.

- **ln-inc_min** The minimum, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.
- **ln-inc_max** The maximum, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.
- **int-rate_min** The minimum, for the current risk category, of the interest rate on the loan.
- **int-rate_max** The maximum, for the current risk category, of the interest rate on the loan.
- **open-acc_min** The minimum, for the current risk category, of the number of open credit lines in the borrower’s credit file.
- **open-acc_max** The maximum, for the current risk category, of the number of open credit lines in the borrower’s credit file.
- **total-acc_min** The minimum, for the current risk category, of the total number of credit lines currently in the borrower’s credit file.
- **total-acc_max** The maximum, for the current risk category, of the total number of credit lines currently in the borrower’s credit file.

Source

https://www.kaggle.com/wendykan/lending-club-loan-data
Loans by risk levels: ten and ninety per cent quantiles Data Set

Description

This data set consist of the ten and ninety per cent quantiles of the intervals for four interval characteristics for 35 risk levels (from A1 to G5) of loans. The original microdata is available at the Kaggle Data Science platform and consists of 887,383 loan records characterized by 75 descriptors. Among the large set of variables available, we focus on borrowers’ income and account and loan information aggregated by the 35 risk levels which are considered as the units of interest.

Usage

data(LoansbyRiskLvs_qntlDt)

Format

A data frame containing 35 observations on the following 8 variables.

- **ln-inc_q0.10** The ten percent quantile, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.
- **ln-inc_q0.90** The ninety percent quantile, for the current risk category, of natural logarithm of the self-reported annual income provided by the borrower during registration.
- **int-rate_q0.10** The ten percent quantile, for the current risk category, of the interest rate on the loan.
- **int-rate_q0.90** The ninety percent quantile, for the current risk category, of the interest rate on the loan.
- **open-acc_q0.10** The ten percent quantile, for the current risk category, of the number of open credit lines in the borrower’s credit file.
- **open-acc_q0.90** The ninety percent quantile, for the current risk category, of the number of open credit lines in the borrower’s credit file.
- **total-acc_q0.10** The ten percent quantile, for the current risk category, of the total number of credit lines currently in the borrower’s credit file.
- **total-acc_q0.90** The ninety percent quantile, for the current risk category, of the total number of credit lines currently in the borrower’s credit file.

Source

https://www.kaggle.com/wendykan/lending-club-loan-data
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 distributions 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.
MANOVA-methods

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, OptCtrl=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” (heterocedastic) 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.
- **OptCtrl**: 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 class 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 geographical regions results =\n")
print(ManvChina)

# Heterocedastic MANOVA tests
HetManvChina <- MANOVA(ChinaT, ChinaTemp$GeoReg, Mxt="Het", CovCase=2:4)
cat("China, heterocedastic MANOVA by geographical 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 geographical 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 geographical 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","NrmndSKN"), CovCase=1:4,
SelCrit=c("BIC","AIC"), OptCtrl=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.

OptCtrl 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


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 likelihood estimation results =\n")
print(ChinaE)
cat("Standard Errors of Estimators:\n")
print(stdEr(ChinaE))
Description

qda performs quadratic discriminant analysis of Interval Data based on classic estimates of a mixture of Gaussian models.

Usage

```r
## 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 'IdtHetMANOVA'
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, ... )
```

Arguments

- `x` An object of class `IData`, `IdtMxtNDE`, `IdtHetMANOVA` or `IdtGenNSNMANOVA` with either the original Interval Data, and estimate of a mixture of gaussian models for Interval Data, or the results of a Interval Data heterocedastic 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 whether a warning message should be printed if the method fails.
Selected model from a list of candidate models saved in object x.

Other named arguments.

References


See Also

`lda`, `snda`, `Roblda`, `Robqda`, `IData`, `IdtMxtNDE`, `IdtHetMANOVA`, `IdtGenNSMANOVA`

Examples

```r
# 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"))

# Quadratic Discriminant Analysis
ChinaT.qda <- qda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- qda discriminant analysis results:
Not run:
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"])
glerrors <-
  apply(CVqda[,,"Nk"]*CVqda[,,"Clerr"],1,sum)/apply(CVqda[,,"Nk"],1,sum)
cat("Average global classification error =",mean(glerrors),"\n")

# End(Not run)
```

---

**qHardRoqF**

*Hardin and Rocke F-quantiles*

Description

p-quantiles of the Hardin and Rocke (2005) scaled F distribution for squared Mahalanobis distances based on raw MCD covariance estimators
Usage

\[ qHardRoqF(p, \text{nobs}, \text{nvar}, h=\text{floor}((\text{nobs}+\text{nvar}+1)/2), \text{adj=TRUE}, \text{lower.tail=TRUE}, \text{log.p=FALSE}) \]

Arguments

- **p**: Vector of probabilities.
- **nobs**: Number of observations used in the computation of the raw MCD Mahalanobis squared distances.
- **nvar**: Number of variables used in the computation of the raw MCD Mahalanobis squared distances.
- **h**: Number of observations kept in the computation of the raw MCD estimate.
- **adj**: logical; if TRUE (default) returns the quantile of the adjusted distribution. Otherwise returns the quantile of the asymptotic distribution.
- **lower.tail**: logical; if TRUE (default), probabilities are \( P(X \leq x) \) otherwise, \( P(X > x) \)
- **log.p**: logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).

Value

The quantile of the appropriate scaled F distribution.

References


See Also

- fasttle, fulltle

---

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, parmstderr=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 lbond 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**: Boolean flag indicating whether a numerically evaluated hessian matrix at the optimum should be computed and returned. Not available for the “nlminb” method.
- **parmstder**: Boolean flag indicating whether parameter asymptotic 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 minimum 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’. Subsequent 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 controlling 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 repetitions of the same minimum objective value, before RepLOptim is stopped 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 stopped 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 stopped 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” method, 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 default 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 asymptotic 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

Description
Roblda and Robqda perform linear and quadratic discriminant analysis of Interval Data based on robust estimates of location and scatter.

Usage

```r
## 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(), ... )
```

Arguments

- **x**: An object of class `IData` with the original Interval Data.
- **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.
- **egvtol**: Tolerance level for the eigenvalues of the product of the inverse within by the between covariance matrices. When an eigenvalue has an absolute value below egvtol, it is considered to be zero.
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 whether 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 l_1 medians. See Todorov and Filzmoser (2009) 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.

... Other named arguments.

References


See Also

lda, qda, snda, IDa, RobEstControl

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"))

# Robust Linear Discriminant Analysis

## Not run:

ChinaT.rlda <- Roblda(ChinaT,ChinaTemp$GeoReg)
cat("Temperatures of China -- robust lda discriminant analysis results:

print(ChinaT.rlda)
Cestimate error rates by tenfold cross-validation with 5 replications

CVlda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=Roblda,CovCase=CovCase(ChinaT.rlda), CVrep=5)
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
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)
**Arguments**

**alpha**
Numeric parameter controlling the size of the subsets over which the trimmed likelihood is maximized; roughly alpha*nrow(Idt) 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 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 the value of argument ‘alpha’ is used directly.

**rawMD2Dist**
The assumed reference distribution of the raw MCD squared distances, which is used to find to cutoffs defining the observations kept in one-step reweighted MCD estimates. Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual Qui-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005).

**MD2Dist**
The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual Qui-squared, the Beta and F distributions proposed by Cerioli (2010).

**eta**
Nominal size of the null hypothesis that a given observation is not an outlier. Defines the raw MCD Mahalanobis distances cutoff used to choose the observations kept in the reweighting step.

**multiCmpCor**
Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at ‘eta’. ‘always’ – testing all n entities at 1.- (1.-'eta'^(1/n)); and ‘iterstep’ – as suggested by Cerioli (2010), make an initial set of tests using the nominal size 1.- (1.-'eta'^(1/n)), and if no outliers were detected stop. Otherwise, make a second step testing for outliers at ‘eta’.

**getdblstar**
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 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 MidP-points, or “LogR” if outliers are only present in LogRanges.
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 'kd-blstar' observations at random, and “Poolm” that divides the original sample into ‘m’ non-overlapping subsets, applies the ‘simple trial’ and the refinement methods to each one of them, and merges the results into a trial subset.

Number of non-overlapping subsets used by the trial method when the argument of ‘trialmethod’ is set to ‘Poolm’.

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.

The amount of output returned by fasttle. Current options are “SetMD2andEst” (default) which returns an ‘IdtSngNDRE’ object with the fasttle estimates, a vector with the final trimmed subset elements used to compute these estimates and the corresponding robust squared Mahalanobis distances, and “SetMD2EstandPrfSt” which returns an ‘IdtSngNDRE’ object with the previous slots plus a list of some performance statistics concerning the algorithm execution.

A RobEstControl object

References


See Also

RobEstControl, fasttle, RobMxtDEst, Roblda, Robqda
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*nrow(Idt) 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**: Inherited from class "CovControlMcd". Number of subsets used for initial estimates.
- **scalefn**: Inherited from class "CovControlMcd" and not used in the package ‘Maint.Data.’
- **maxcsteps**: Inherited from class "CovControlMcd" and not used in the package ‘Maint.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.
- **rawMD2Dist**: The assumed reference distribution of the raw MCD squared distances, which is used to find to cutoffs defining the observations kept in one-step reweighted MCD estimates. Alternatives are ‘ChiSq’, ‘HardRockeAsF’ and ‘HardRockeAdjF’, respectively for the usual Qui-squared, and the asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005).
- **MD2Dist**: The assumed reference distributions used to find cutoffs defining the observations assumed as outliers. Alternatives are “ChiSq” and “CerioliBetaF” respectively for the usual Qui-squared, and the Beta and F distributions proposed by Cerioli (2010).
**exts**: Nominal size of the null hypothesis that a given observation is not an outlier. Defines the raw MCD Mahalanobis distances cutoff used to choose the observations kept in the reweightening step.

**multicmpcor**: Whether a multicomparison correction of the nominal size (eta) for the outliers tests should be performed. Alternatives are: ‘never’ – ignoring the multicomparisons and testing all entities at ‘eta’, ‘always’ – testing all n entities at 1.- (1.-‘eta’^((1/n))); and ‘iterstep’ – as sugested by Cerioli (2010), make an initial set of tests using the nominal size 1.- (1.-‘eta’^((1/n))), and if no outliers were detected stop. Otherwise, make a second step testing for outliers at ‘eta’.

**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 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-overlapping subsets, apllies the ‘simple trial’ and the refinement methods to each one of them, and merges the results into a trial subset.

**m**: Number of non-overlapping 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” whichever returns a list with the previous three components plus 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**


RobMxtDEst-methods

Methods for Function RobMxtDEst in Package 'MAINT.Data'

Description

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

Usage

```r
## 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” (heterocedastic).
- **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 $l_1$ medians. See Todorov and Filzmoser (2009) 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 (in package pcaPP) used to find the multivariate l_1 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 class IdtMxNDRE, containing the estimation results.

References


See Also
IdtMxNDRE, RobEstControl.

---

**snda-methods**

*Skew-Normal Discriminant Analysis of Interval Data*

**Description**

snda performs discriminant analysis of Interval Data based on estimates of mixtures of Skew-Normal models

**Usage**

```r
## 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 'IdtLocNSMANOVA'
snda( x, prior="proportions",
```
Arguments

x An object of class `IData`, `IdtLocSNMANOVA`, `IdtLocNSNNMANOVA`, `IdtGenSNMANOVA` or `IdtGenSNNSMANOVA` with either the original Interval Data, or the results of an 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.

Mxt Indicates the type of mixing distributions to be considered. Current alternatives are “Loc” (location model – groups differ only on the location parameters of a Skew-Normal model) and “Gen” (general model – groups differ on all parameters of a Skew-Normal models).

silent A boolean flag indicating whether a warning message should be printed if the method fails.

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 an eigenvalue has an absolute value below egvtol, it is considered to be zero.

... Other named arguments.

References


### See Also

`lda`, `qda`, `Roblda`, `Robqda`, `IData`, `IdtLocSNMANOVA`, `IdtLocNSNMANOVA`, `IdtGenSNMANOVA`, `IdtGenSNMANOVA`

### Examples

```r
## Not run:

# 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"))

# Skew-Normal based discriminant analysis, assuming that the different regions differ
# only in location parameters
ChinaT.locsnda <- snda(ChinaT,ChinaTemp$GeoReg,Mxt="Loc")

cat("Temperatures of China -- SkewNormal location model discriminant analysis results:\n")
print(ChinaT.locsnda)

#Estimate error rates by three-fold cross-validation without replication
CVlocsnda <- DACrossVal(ChinaT,ChinaTemp$GeoReg,TrainAlg=snda,Mxt="Loc",
        CovCase=CovCase(ChinaT.locsnda),kfold=3,CVrep=1)

summary(CVlocsnda[,"Clerr"])

glerrors <-
   apply(CVlocsnda[,"Nk"]*CVlocsnda[,"Clerr"],1,sum)/apply(CVlocsnda[,"Nk"],1,sum)

cat("Average global classification error =",mean(glerrors),"\n")

# Skew-Normal based discriminant analysis, assuming that the different regions may differ
# in all SkewNormal parameters
ChinaT.gensnda <- snda(ChinaT,ChinaTemp$GeoReg,Mxt="Gen")

cat("Temperatures of China -- SkewNormal general model discriminant analysis results:\n")
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)
```
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

```r
## 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`
summary-methods

Methods for Function summary in Package ‘MAINT.Data’

Description

summary methods for classes defined in Package ‘MAINT.Data’.

Usage

```r
## S4 method for signature 'IdtMclust'
summary(object, parameters = FALSE, classification = FALSE, ...)
```

Arguments

- `object`: An object of class `IdtMclust` representing the results of fitting Gaussian mixtures to interval data objects.
- `parameters`: A boolean flag indicating if the parameter estimates of the optimal mixture should be displayed.
- `classification`: A boolean flag indicating if the crisp classification resulting from the optimal mixture should be displayed.
- `...`: Other named arguments.

---

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

```r
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 being evaluated in the NULL hypothesis.
- `FullMod`: Either indices of the general models being 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 class ConfTests with the results of the tests performed

Examples

```r
# 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[,1:4])
cat("China maximum likelihood 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)
```

Description

S4 methods for function var. These methods extract estimates of variance-covariance matrices for
the models fitted to Interval Data.

Usage

```r
## S4 method for signature 'IdtNDE'
var(x)
## S4 method for signature 'IdtSNDE'
var(x)
## S4 method for signature 'IdtNandSNDE'
var(x)
## S4 method for signature 'IdtMxNDE'
var(x)
## S4 method for signature 'IdtMxSNDE'
var(x)
```
vcov—methods

Arguments

x

An object representing a model fitted to interval data.

Value

For the `IdtNDE`, `IdtSNDE` and `IdtNandSNDE` methods or `IdtMxNDE`, `IdtMxSNDE` methods with slot “Hmcdt” equal to TRUE: a matrix with the estimated covariances.

For the `IdtMxNDE`, and `IdtMxSNDE` methods with slot “Hmcdt” equal to FALSE: a three-dimensional array with a matrix with the estimated covariances for each group at each level of the third dimension.

See Also

`cor`

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

```r
## 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>An object representing a model fitted to interval data.</td>
</tr>
<tr>
<td>selmodel</td>
<td>Selected model from a list of candidate models saved in object.</td>
</tr>
<tr>
<td>group</td>
<td>The 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. Note that this argument is only used in heterocedastic models, i.e. in the <code>IdtMxNDE</code>, <code>IdtMxSNDE</code> methods when the object slot “Hmcdt” is set to to FALSE.</td>
</tr>
</tbody>
</table>
Additional arguments for method functions.

Value

For the `IdtNDE`, `IdtSNDE` and `IdtNandSNDE` methods or `IdtMxNDE`, `IdtMxSNDE` methods with slot “Hmcdt” equal to TRUE: a matrix of the estimated covariances between the parameter estimates. For the `IdtMxNDE`, and `IdtMxSNDE` methods with slot “Hmcdt” equal to FALSE: 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

*Topic AIC
   BestModel-methods, 7
*Topic BIC
   BestModel-methods, 7
*Topic Discriminant Analysis Interval Data
   lda-methods, 51
cor-methods, 60
Robda-methods, 65
qda-methods, 60
snda-methods, 73
*Topic EM algorithm
   EMControl-class, 14
*Topic Interval Data
   coef-methods, 9
cor-methods, 11
IdtNandSNDE-class, 40
IdtNDE-class, 41
IdtSNDE-class, 42
Ida-methods, 51
MAINT.Data-package, 3
qda-methods, 60
Robda-methods, 65
snda-methods, 73
stdErr-methods, 76
var-methods, 78
vcov-methods, 79
*Topic MAINT.Data
   MAINT.Data-package, 3
*Topic MANOVA for Interval Data
   MAINT.Data-package, 3
*Topic MANOVA
   IdtMANOVA-class, 28
MANOVA-methods, 56
*Topic Parametric modelling of Interval Data
   MAINT.Data-package, 3
*Topic RobMxtDEst
   RobMxtDEst-methods, 72
*Topic Statistical tests for Interval Data
   MAINT.Data-package, 3
*Topic Symbolic Data Analysis
   MAINT.Data-package, 3
*Topic classes
   Conftests-class, 10
EMControl-class, 14
IData-class, 24
IdtE-class, 26
Idtlda-class, 27
IdtMANOVA-class, 28
IdtMclust-class, 30
IdtMxE-class, 33
IdtMxNandSNDE-class, 34
IdtMxNDE-class, 35
IdtMxNDRE-class, 37
IdtMxSNDE-class, 38
IdtMxtNDE-class, 40
IdtNandSNDE-class, 40
IdtNDE-class, 41
Idtqda-class, 41
IdtSNDE-class, 42
IdtSNgenda-class, 43
IdtSngNandSNDE-class, 44
IdtSngNDE-class, 45
IdtSngNDRE-class, 47
IdtSngSNDE-class, 48
IdtSNlocda-class, 50
LRTest-class, 56
RobEstControl-class, 70
*Topic coef
   coef-methods, 9
*Topic cor
   cor-methods, 11
*Topic cross-validation
   DACrossVal, 12
*Topic datasets
   Cars, 8
   ChinaTemp, 8
coef, IdtNandSNDE-method (coef-methods), 9
coef, IdtNDE-method (coef-methods), 9
coef, IdtSNDE-method (coef-methods), 9
coef-methods (coef-methods), 9
ConfTests, 56
ConfTests-class, 10
cor, 79
cor (cor-methods), 11
cor, IdtMxNDE-method (cor-methods), 11
cor, IdtMxSNDE-method (cor-methods), 11
cor, IdtNandSNDE-method (cor-methods), 11
cor, IdtNDE-method (cor-methods), 11
cor, IdtSNDE-method (cor-methods), 11
cor-methods (cor-methods), 11
CovCase, Idtlda-method (Idtlda-class), 27
CovCase, Idtqda-method (Idtqda-class), 41
CovCase, IdtSNGenA-method (IdtSNGenA-class), 43
CovCase, IdtSNlocda-method (IdtSNlocda-class), 50
CovControl, 71
CovControlMcd, 71
DACrossVal, 12
EMControl, 14, 14, 15, 32
EMControl-class, 14
extmatrix-class, 15
fasttle, 21, 23, 26, 27, 40, 41, 45, 48, 62, 67, 69, 72
fasttle (fasttle-methods), 15
fasttle, IData-method (fasttle-methods), 15
fasttle-methods, 15
fulltle, 18, 23, 26, 27, 40, 41, 45, 48, 62
fulltle (fulltle-methods), 19
fulltle, IData-method (fulltle-methods), 19
fulltle-methods, 19
getIdtOutl, 18, 21, 22
H1res (MANOVA-methods), 56
H1res, IdtMANOVA-method (MANOVA-methods), 56
head, IData-method (IData-class), 24
IData-class, 24
IdtClMANOVA, 28, 51, 52
IdtClMANOVA-class (IdtMANOVA-class), 28
IdtE, 7, 17, 20, 33–40, 44–49, 77
IdtE-class, 26
IdtGenNSNMANOVA, 28, 60, 61, 74
IdtGenNSNMANOVA-class (IdtMANOVA-class), 28
IdtGenNSNMANOVA, 28, 74, 75
IdtGenNSNMANOVA-class (IdtMANOVA-class), 28
IdtHetNMANOVA, 28, 60, 61
IdtHetNMANOVA-class (IdtMANOVA-class), 28
Idtlda-class, 27
IdtLocSNMANOVA, 28, 51, 52, 74, 75
IdtLocSNMANOVA-class (IdtMANOVA-class), 28
IdtMANOVA, 28
IdtMANOVA-class, 28
IdtMclust, 32, 77
IdtMclust, 14, 31
IdtMclust (IdtMclust-methods), 31
IdtMclust, IData-method (IdtMclust-methods), 31
IdtMclust-class, 30
IdtMclust-methods, 31
IdtMxE, 34–40
IdtMxE-class, 33
IdtMxNandSNDE, 40, 45
IdtMxNandSNDE-class, 34
IdtMxNDE, 11, 36, 38, 40, 41, 46, 79, 80
IdtMxNDE-class, 35
IdtMxNDRE, 36, 40, 41, 48, 73
IdtMxNDRE-class, 37
IdtMxSNDE, 11, 39, 42, 43, 49, 79, 80
IdtMxSNDE-class, 38
IdtMxtNDE, 51, 52, 60, 61
IdtMxtNDE-class, 40
IdtNandSNDE, 9, 11, 79, 80
IdtNandSNDE-class, 40
IdtNDE, 11, 43, 79, 80
IdtNDE-class, 41
Idtqda-class, 41
IdtSNDE, 9, 11, 41, 79, 80
IdtSNDE-class, 42
IdtSngDE, 34, 45, 46, 48, 49
IdtSngDE-class (IdtE-class), 26
IdtSNGenda-class, 43
IdtSngNandSNDE, 35, 40
IdtSngNandSNDE-class, 44
IdtSngNDE, 22, 36, 41, 48, 49
IdtSngNDE-class, 45
IdtSngNDRD-class, 18, 22, 38, 41, 46
IdtSngNDRE-class, 47
IdtSngSNE, 40, 42, 43, 46
IdtSngSNDE-class, 48
IdtSNlocda-class, 50
is.IData (IData), 23
lda, 13, 26, 30, 61, 66, 75
lda (lda-methods), 51
lda, IData-method (lda-methods), 51
lda, IdtCLMANOVA-method (lda-methods), 51
lda, IdtLocNSMANOVA-method (lda-methods), 51
lda, IdtMxtNDE-method (lda-methods), 51
lda-methods, 51
LoansbyPurpose_minmaxDt, 53
LoansbyRiskLvs_minmaxDt, 54
LoansbyRiskLvs_qntLtDt, 55
LRTest, 10, 28, 29
LRTest-class, 56
MAINT.Data (MAINT.Data-package), 3
MAINT.Data-package, 3
MANOVA, 26–28, 30, 34–36, 40–44, 51, 56
MANOVA (MANOVA-methods), 56
MANOVA, IData-method (MANOVA-methods), 56
MANOVA-methods, 56
matrix, 15
mle, 10, 26, 27, 40, 41, 43, 45, 46, 49, 56
mle (mle-methods), 58
mle, IData-method (mle-methods), 58
mle-methods, 58
predict, Idtllda-method (Idtllda-class), 27
predict, Idtqda-method (Idtqda-class), 41
predict, IdtSNgenda-method (IdtSNgenda-class), 43
predict, IdtSNlocda-method (IdtSNlocda-class), 50
print, IData-method (IData-class), 24
qda, 13, 26, 28, 30, 42, 52, 66, 75
qda (qda-methods), 60
qda, IData-method (qda-methods), 60
qda, IdtGenNSMANOVA-method (qda-methods), 60
qda, IdtHetNSMANOVA-method (qda-methods), 60
qda, IdtMxtNDE-method (qda-methods), 60
qda-methods, 60
qHardRoqF, 61
RepLOptim, 57, 59, 62
Robda-methods, 65
RobEstControl, 17, 18, 66, 67, 69, 72, 73
RobEstControl-class, 70
Roblda, 26, 28, 52, 61, 67, 69, 72, 75
Roblda (Robda-methods), 65
Roblda, IData-method (Robda-methods), 65
Roblda-methods (Robda-methods), 65
RobMxtDEst, 26, 27, 34, 35, 38, 40, 41, 67, 69, 72
RobMxtDEst (RobMxtDEst-methods), 72
RobMxtDEst, IData-method (RobMxtDEst-methods), 72
RobMxtDEst-methods, 72
Robqda, 26, 28, 42, 52, 61, 67, 69, 72, 75
Robqda (Robda-methods), 65
Robqda, IData-method (Robda-methods), 65
Robqda-methods (Robda-methods), 65
rrcov.control, 16
show, Conftests-method (Conftests-class), 10
show, IdtE-method (IdtE-class), 26
show, Idtllda-method (Idtllda-class), 27
show, IdtMANOVA-method (MANOVA-methods), 56
show, IdtMclust-method (IdtMclust-class), 30
show, Idtqda-method (Idtqda-class), 41
show, IdtSNgenda-method (IdtSNgenda-class), 43
INDEX

show, IdtSNlocda-method
( IdtSNlocda-class), 50
show, LRTest-method (LRTest-class), 56
snd a, 28, 30, 44, 51, 52, 61, 66
snd a (snd a-methods), 73
snd a, IData-method (snd a-methods), 73
snd a, IdtGenNSNMANOVA-method
(snda-methods), 73
snd a, IdtGenSNMANOVA-method
(snda-methods), 73
snd a, IdtLocNSNMANOVA-method
(snda-methods), 73
snd a, IdtLocNSNMANOVA-method
(snda-methods), 73
snd a-methods, 73
stdEr, 9, 80
stdEr (stdEr-methods), 76
stdEr, IdtNandSNDEN-method
(stdEr-methods), 76
stdEr, IdtNE-method (stdEr-methods), 76
stdEr, IdtSNDE-method (stdEr-methods), 76
stdEr-methods, 76
stdEr-methods (stdEr-methods), 76
summary, IData-method (IData-class), 24
summary, IdtMclust-method
(summary-methods), 77
summary-methods, 77
tail, IData-method (IData-class), 24
testMod (testMod-methods), 77
testMod, IdtE-method (IdtE-class), 26
testMod-methods, 77
var, 11
var (var-methods), 78
var, IdtMxNDE-method (var-methods), 78
var, IdtMxSNDE-method (var-methods), 78
var, IdtNandSNDEN-method (var-methods), 78
var, IdtNDE-method (var-methods), 78
var, IdtSNDE-method (var-methods), 78
var-methods, 78
var-methods (var-methods), 78
vcov, 9, 76
vcov (vcov-methods), 79
vcov, IdtMxNDE-method (vcov-methods), 79
vcov, IdtMxSNDE-method (vcov-methods), 79
vcov, IdtNandSNDEN-method
(vcovi-methods), 79
vcov, IdtNDE-method (vcov-methods), 79
vcov, IdtSNDE-method (vcov-methods), 79
vcov-methods, 79
vcov-methods (vcov-methods), 79