Package ‘hisemi’

July 9, 2017

Type Package
Title Hierarchical Semiparametric Regression of Test Statistics
Version 1.1-0
Date 2017-07-09
Author Long Qu
Maintainer Long Qu <rtistician@gmail.com>
Depends R (>= 2.12)
Imports Matrix , Iso(>= 0.0-5), splines, fda, stats, graphics, methods
Suggests multtest, qvalue, pi0
Description Methods for hierarchical semiparametric regression models for test statistics are implemented in this package. Specifically, test statistics given the null/alternative hypotheses are modeled parametrically, whereas the unobservable status of null/alternative hypotheses are modeled using nonparametric additive logistic regression over covariates.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2017-07-09 06:22:30 UTC

R topics documented:

  hisemi-package ................................................. 2
  coef.hisemit .................................................. 4
  confint.hisemit ............................................... 5
  directSum ..................................................... 6
  EMupdate ...................................................... 7
  fitted.hisemit ............................................... 8
  logistic.enp .................................................. 9
  logit ........................................................ 10
  logLik.hisemit ............................................... 11
  n.knots ....................................................... 12
  NRupdate ..................................................... 13
hisemi-package

Hierarchical semiparametric regression model to a large number of parametric test statistics

Description

The package includes functions for fitting hierarchical semiparametric regression model to a large number of parametric test statistics.

Details

- **Package**: hisemi
- **Type**: Package
- **Version**: 1.1-0
- **Date**: 2017-07-09
- **License**: GPL version 2 or newer
- **LazyLoad**: yes

- **penLik.EMNewton** is the major interface function to be called.
- **coef.hisemit** extracts the estimated parameters.
- **confint.hisemit** returns the confidence intervals.
- **directSum** computes the direct sum of matrices.
- **EMupdate** is the EM algorithm used in **penLik.EMNewton**.
- **fitted.hisemit** extracts the fitted values.
- **logLik.hisemit** returns the log likelihood.
- **NRupdate** is the Newton-Raphson algorithm used in **penLik.EMNewton**.
- **plot.hisemit** plots the fitted model.
- **print.hisemit** print summary information.
- **residuals.hisemit** returns the residuals.
• `scaledTMix.null` fits the null model with common \( \pi_0 \).
• `scaledTMix.psat` fits the partially saturated model with free \( \pi_0 \) and common scale factor.
• `scaledTMix.sat` fits the completely saturated model with free \( \pi_0 \) and free scale factor.
• `vcov.hisemit` returns the sandwich variance-covariance matrix.

Author(s)

Long Qu <rtistician@gmail.com>
Maintainer: Long Qu <rtistician@gmail.com>

References


See Also

`pi0-package`

Examples

```r
## simulate some fake data
G=100  ## for demonstration only. Normally, G should be much larger
sdncp=1.3
n1=n2=5
df=n1+n2-2
set.seed(54457704)
x=runif(G,1,G)
f=function(x)sin(x*pi/1000)+1
Pi.i=1/(1+exp(f(x)))
Z.i=rbinom(G,1,1-Pi.i)
t0.i=rt(G,df)
ncp.i=rnorm(G,0,sdncp)
t1.i=rt(G,df,ncp.i)
t.i=ifelse(Z.i==0,t0.i,t1.i)

## fit model
(plfit=penLik.EMNewton(t.i, x, df, spar=10^seq(0,8,length=30),plotit=FALSE))
(plfit0=scaledTMix.null(t.i, df))

## Not run:
plot(plfit)
plot(t.i, plfit$lfdr, pch='.'
lines(sort(t.i), plfit0$lfdr[order(t.i)], col=2, lwd=3)

## End(Not run)
```
Extracts fitted parameters from a hisemit object

### Description

Extracts fitted parameters from a hisemit object

### Usage

```r
## S3 method for class 'hisemit'
coef(object, scale.parameterization = c("r",
    "scale.factor", "sd.ncp"), ...)
```

### Arguments

- **object**: A hisemit object
- **scale.parameterization**: One of "r", "scale.factor", "sd.ncp". See details.
- **...**: Not used.

### Details

For the scale parameter, there are three parameterizations.

- 'scale.factor' means the multiplicative scaling factor (greater than 1).
- 'sd.ncp' means the equivalent standard deviation of the noncentrality parameters.
- 'r' means \(\log(\text{scale.factor}-1)\), which is in the range of whole real line.

### Value

A numeric vector of estimated parameters

### Author(s)

Long Qu <rtistician@gmail.com>

### References


### See Also

- `plot.hisemit`, `fitted.hisemit`, `coef.hisemit`, `vcov.hisemit`, `residuals.hisemit`, `logLik.hisemit`, `confint.hisemit`, `print.hisemit`, `penLik.EMNewton`
confint.hisemit

Extract Wald-type asymptotic confidence intervals from a hisemit object

Description

Extract Wald-type asymptotic confidence intervals from a hisemit object.

Usage

```r
## S3 method for class 'hisemit'
confint(object, parm = c("lfdr", "fpp", "beta", "scale.fact", "sd.ncp", "r", "coef", "pi0", "f"), level = 0.95, component, ...)
```

Arguments

- `object`: A hisemit object.
- `level`: A numeric scalar between 0 and 1, specifying the level of confidence.
- `component`: Specifying which additive component to be extracted. See details.
- `...`: Currently not used.

Details

The `parm` could be:

- 'lfdr' Local false discovery rates
- 'fpp' False positive proportions; not implemented yet.
- 'beta' Regression coefficients
- 'scale.fact' Multiplicative scale factor
- 'r' \( \log(\text{scale.fact}-1) \)
- 'sd.ncp' Equivalent standard deviation of noncentrality parameters
- 'coef' All parameters
- 'pi0' Prior probability of true null hypotheses
- 'f' Underlying smooth function

The `component` specifies which component of the fitted smooth function to be extracted. If `component` is missing, the overal function is returned. If `component` is numeric, then it specifies the smooth function corresponding to which covariate to be extracted.
**Value**

A matrix (or vector) with columns giving lower and upper confidence limits for each parameter.

**Author(s)**

Long Qu <rtistician@gmail.com>

**References**


**See Also**

plot.hisemit, fitted.hisemit, coef.hisemit, vcov.hisemit, residuals.hisemit, logLik.hisemit, confint.hisemit, print.hisemit, penLik.EMNewton

---

**directSum**

**Direct sum of matrices**

**Description**

Direct sum of matrices, i.e., put matrices along the diagonal

**Usage**

directSum(...)

**Arguments**

... matrices to be directSumed.

**Details**

The function treats each matrix as an element and put the element along the diagonal line.

**Value**

A matrix, which is the direct sum of ....

**Author(s)**

Long Qu <rtistician@gmail.com>

**Examples**

```r
A=matrix(1:9,3,3)
all.equal(directSum(A), A) #TRUE
(B=directSum(-1,A))
directSum(A,B)
```
**EMupdate**  
*Utility function performing EM algorithm updates*

**Description**

Utility function performing EM algorithm updates for `penLik.EMNewton`

**Usage**

```r
EMupdate(starts, nLogLik.pen, optim.method, H, tstat, df, dt0,
         spar.Pen.mat, em.iter.max = 10, em.beta.iter.max = 1,
         scale.conv = 0.001, lfdr.conv = 0.001,
         NPLL.conv = 0.001, debugging = FALSE)
```

**Arguments**

- **starts**: A numeric vector of starting values, in 'r' parameterization of the `scale.fact` parameter.
- **nLogLik.pen**: A function computing negative penalized log likelihood.
- **optim.method**: One of BFGS, CG, L-BFGS-B, Nelder-Mead, SANN, nlminb, NR, the method used for optimization.
- **H**: Design matrix.
- **tstat**: A numeric vector of t-statistics.
- **df**: A numeric scalar or vector of degrees of freedom.
- **dt0**: A numeric vector of the central t-density evaluated at the t-statistics.
- **spar.Pen.mat**: Smoothing parameter times the penalty matrix.
- **em.iter.max**: Maximum number of EM iterations.
- **em.beta.iter.max**: Maximum number of iterations in maximization step with respect to regression coefficients.
- **scale.conv**: A small numeric scalar specifying the convergence criterion for the scale parameter.
- **lfdr.conv**: A small numeric scalar specifying the convergence criterion for the local false discovery rates.
- **NPLL.conv**: A small numeric scalar specifying the convergence criterion for the negative penalized log likelihood.
- **debugging**: A logical scalar indicating whether debugging mode of the code should be run.

**Value**

A numeric vector of updated parameter estimates. The scale factor is in the \( \log(\text{scale.fact}-1) \) parameterization.
Author(s)
Long Qu <rtistician@gmail.com>

References

See Also
penLik.EMNewton, NRupdate

---

fitted.hisemit

**Extract fitted values from a hisemit object**

**Description**
Extract fitted values from a hisemit object

**Usage**
```r
## S3 method for class 'hisemit'
fitted(object, fitted.type = c("lfdr", "fpp", "pi0", "f"),
        gene.list, component, ...)
```

**Arguments**
- `object`: A hisemit object
- `fitted.type`: One of c("lfdr", "fpp", "pi0", "f"). See details.
- `gene.list`: Not implemented
- `component`: Specifying which additive component is extracted. See details.
- `...`: Not used.

**Details**
For the `fitted.type` argument,
- 'lfdr' specifies local false discovery rates to be extracted.
- 'fpp' specifies false positive proportion to be extracted.
- 'pi0' specifies prior probability of null hypothesis to be extracted.
- 'f' specifies the smooth function to be extracted.
When fitted.type='f', the component specifies which component of the fitted smooth function to be extracted.
If component is missing, the overall function is returned.
If component='intercept', only the intercept term is returned.
If component is numeric, then it specifies the smooth function corresponding to which covariate to be extracted.

Value
A numeric vector

Author(s)
Long Qu <rtistician@gmail.com>

References

See Also
plot.hisemit, fitted.hisemit, coef.hisemit, vcov.hisemit, residuals.hisemit, logLik.hisemit, confint.hisemit, print.hisemit, penLik, EMNewton

logistic.enp

*Fit a logistic curve to the raw effective number of parameters over log smoothing parameter*

**Description**
Fit a logistic curve to the raw effective number of parameters over log smoothing parameter

**Usage**
logistic.enp(log.spar, enps, maximum, minimum = 2, eps = 1e-08)

**Arguments**
- `log.spar`: A numeric vector of log smoothing parameters
- `enps`: A numeric vector of raw effective number of parameters
- `maximum`: A numeric scalar of upper bound of the effective number of parameters
- `minimum`: A numeric scalar of lower bound of the effective number of parameters
- `eps`: A small numeric scalar of error of tolerance
Details
The purpose is of this function is to reduce some estimation variability of effective number of parameters. The function heuristically fit a smooth logistic curve to the raw effective number of parameters.

Value
A numeric vector of fitted effective number of parameters, with attributes

- log.spar: log of smoothing parameter
- rate: rate parameter of the logistic curve
- mdis: the center parameter of the logistic curve
- pow: the exponent parameter of the logistic curve
- fit: a nonlinear least squared fit object
- goodenp.idx: Indices of enps whose values are reasonably estimated
- mode: The model of enps from a isotonic fit

Author(s)
Long Qu <rtistician@gmail.com>

See Also
nls

logit
Logit link and its inverse

Description
Logit link and its inverse

Usage
logit(mu)
logit.inv(eta)

Arguments
mu See make.link.
eta See make.link.

Details
logit is defined as make.link("logit")$linkfun.
logit.inv is defined as make.link("logit")$linkinv.
logLik.hisemit

Value
A link function or its inverse. See make.link.

Author(s)
Long Qu <rtistician@gmail.com>

See Also
make.link

logLik.hisemit  
Extract the log likelihood from a hisemit object

Description
Extract the log likelihood from a hisemit object.

Usage
## S3 method for class 'hisemit'
logLik(object, take.sum = TRUE,...)

Arguments
object  
A hisemit object
take.sum  
A logical scalar, indicating whether total log likelihood or the log likelihood of each data point should be extracted.
...  
Currently not used

Value
An object of class 'logLik'.

Author(s)
Long Qu <rtistician@gmail.com>

References

See Also
plot.hisemit, fitted.hisemit, coef.hisemit, vcov.hisemit, residuals.hisemit, logLik.hisemit, confint.hisemit, print.hisemit, penLik.EMNewton
n.knots

<table>
<thead>
<tr>
<th>n.knots</th>
<th>Number of spline knots</th>
</tr>
</thead>
</table>

**Description**

Compute the number of spline knots to be the largest integer not greater than \( \min(cutoff, \max(0, n-cutoff)^{rate}) \).

**Usage**

\[
n.knots(n, cutoff=35, rate=0.2)
\]

**Arguments**

- **n**: An integer vector of sample sizes.
- **cutoff**: A numeric vector of cutoff values.
- **rate**: A numeric vector of rates at which the number of knots increases with the sample size.

**Value**

An integer vector, being the number of knots for splines.

**Author(s)**

Long Qu <rtistician@gmail.com>

**References**


**Examples**

\[
n.knots(10, 35, .2)
n.knots(35, 35, .2)
n.knots(135, 35, .2)
n.knots(1350, 35, .2)
n.knots(13500, 35, .2)
\]
NRupdate

**Utility function performing Newton-Raphson algorithm updates**

**Description**

Utility function performing Newton-Raphson algorithm updates for penLik.EMNewton

**Usage**

```r	nRupdate(f, starts, gradient, hessian, ..., ridge0 = 1e-06, 
tolerance = sqrt(.Machine$double.eps), 
iternax = 1500, halving.max = Inf, relative = FALSE, 
return.hessian = FALSE, debugging = FALSE)
```

**Arguments**

- `f`: Objective function to be minimized
- `starts`: A numeric vector of starting values
- `gradient`: The gradient function of `f`
- `hessian`: The Hessian function of `f`
- `...`: Additional arguments to be passed to `f`
- `ridge0`: A small ridge factor; obsolete. Current version uses `nearPD` to stabilize hessian
- `tolerance`: A small numeric scalar giving the convergence criterion
- `iter.max`: Maximum number of iterations
- `halving.max`: Maximum number of step-halfing
- `relative`: A logical scalar indicating if relative convergence should be checked.
- `return.hessian`: A logical scalar indicating if the final Hessian matrix is returned.
- `debugging`: A logical scalar indicating if the debugging mode of the code should be run.

**Value**

A numeric vector of updated parameters, with attributes

- 'objective': The final evaluated objective function
- 'gradient': The final gradient vector
- 'iter': The number of iterations
- 'hessian': The final Hessian matrix, only available if `return.hessian=TRUE`.

**Author(s)**

Long Qu <rtistician@gmail.com>
OsplinePen

References

See Also
penLik.EMNewton, EMupdate

OsplinePen                  O-spline penalty matrix

Description
This function returns the penalty matrix for smoothing spline of any order.

Usage
OsplinePen(Boundary.knots, knots, ord=1)

Arguments
Boundary.knots  A length 2 numeric vector, giving the boundary knot values.
knots         A numeric vector of internal knots.
ord            A numeric integer, which is the order of the derivatives on which squared integral will become the smoothness penalty.

Value
A symmetrix penalty matrix.

Note
When knots are selected by all distinct x values, this returns the penalty matrix of smoothing splines.

Author(s)
Long Qu [<rtistician@gmail.com>]

References

See Also
bsplinepen
penLik.EMNewton

**Examples**

```r
b.k=c(0,1)
br=seq(.1,.9,by=.1)
O1=OsplinePen(b.k, br, 1)
O2=OsplinePen(b.k, br, 2)
O3=OsplinePen(b.k, br, 3)
O4=OsplinePen(b.k, br, 4)
O5=OsplinePen(b.k, br, 5)
O6=OsplinePen(b.k, br, 6)

library(fda)
## Not run:
des1=create.bspline.basis(c(0,1),norder=2, breaks=br)
P1=bsplinepen(des1, 1) # ERROR
max(abs(P1-01))

## End(Not run)
des2=create.bspline.basis(c(0,1),norder=4, breaks=c(b.k[1], br, b.k[2]))
P2=bsplinepen(des2, 2)
max(abs(P2-02))
des3=create.bspline.basis(c(0,1),norder=6, breaks=c(b.k[1], br, b.k[2]))
P3=bsplinepen(des3, 3)
max(abs(P3-03))
des4=create.bspline.basis(c(0,1),norder=8, breaks=c(b.k[1], br, b.k[2]))
P4=bsplinepen(des4, 4, c(0,1))
max(abs((P4-04)/(P4+04)*2),na.rm=TRUE)
des5=create.bspline.basis(c(0,1),norder=10, breaks=c(b.k[1], br, b.k[2]))
P5=bsplinepen(des5, 5, c(0,1))
max(abs((P5-05)/(P5+05)*2),na.rm=TRUE)
des6=create.bspline.basis(c(0,1),norder=12, breaks=c(b.k[1], br, b.k[2]))
P6=bsplinepen(des6, 6, c(0,1))
max(abs((P6-06)/(P6+06)*2),na.rm=TRUE)
```

**penLik.EMNewton**

*Fits hierarchical semiparametric regression model to t-statistics*

**Description**

Fits hierarchical semiparametric regression model to t-statistics

**Usage**

```r
penLik.EMNewton(tstat, x, df, spar = c(10*seq(-1,8,length=30), Inf),
nknots = n.knots(length(tstat)), starts,
```
tuning.method = c("NIC", "CV"), cv.fold = 5, pen.order=1,
poly.degree=pen.order+2-1, optim.method =
c("nlminb", "BFGS", "CG", "L-BFGS-B", "Nelder-Mead", "SANN", "NR"),
logistic.correction = TRUE, em.iter.max = 10,
em.beta.iter.max = 1, newton.iter.max = 1500,
scale.conv = 0.001, lfdr.conv = 0.001, NPLL.conv = 0.001,
debugging = FALSE, plotit = TRUE, ...)

Arguments

tstat A numeric vector t-statistics
x A numeric matrix of covariates, with nrow(x) being length(tstat)
df A numeric scalar or vector of degrees of freedom
spar A numeric vector of smoothing parameter lambda
nknots A numeric scalar of number of knots
starts An optional numeric vector of starting values
tuning.method Either 'NIC' or 'CV', specifying the method to choose the tuning parameter
spar

CV.fold A numeric scalar of the fold for cross-validation. Ignored if tuning.method='NIC'.
pen.order A numeric scalar of the order of derivatives of which squared integration will be
used as roughness penalty.
poly.degree A numeric scalar of the degree of B-splines.
optim.method A character scalar specifying the method of optimization.
logistic.correction A logical scalar specifying whether or not the effective number of parameters
should be corrected using a logistic curve

em.iter.max A numeric scalar specifying the maximum number of EM iterations. If being
Inf, then EM algorithm is used. If being 0, then Newton method is used. Other-
wise, EM algorithm is used initially, followed by Newton method.

em.beta.iter.max A numeric scalar specifying the maximum number of iterations in the maxi-
mization step for the beta parameters in the EM algorithm. If being Inf, the
original EM is used. If being 1 or other numbers, the generalized EM algorithm
is used.

newton.iter.max A numeric scalar specifying the maximum number of iterations in Newton method.
scale.conv A small numeric scalar specifying the convergence criterion for the scale param-
eter.
lfdr.conv A small numeric scalar specifying the convergence criterion for the local false
discovery rates.
NPLL.conv A small numeric scalar specifying the convergence criretion for the negative
penalized log likelihood.
debugging A logical scalar. If TRUE, then dump.frame will be called whenever error occurs.
plotit A logical scalar specifying whether a plot should be generated.
... Currently not used.
Value

An list of class hisemit:

1fdr: A numeric vector of local false discovery rates.

model A list of tstat, df and x, which are the same as arguments

scale.fact: A list with
  • scale.fact: Scale factor
  • sd.ncp: Equivalent standard deviation of noncentrality parameters
  • r: A reparameterization of scale.fact
  • t.cross: \( \sqrt{df*(s^2/(df+1))^{-1} /(1-s^2-2*df/(df+1)))} \)
  where \( s \) is the scale.fact

pi0: A numeric vector of mixing proportions for the central t component

tuning: A list with
  • mean: Mean criterion
  • var: Variance of criterion across observations
  • grp: Cross-validation group membership
  • method: The tuning.method used.
  • final: The minimum mean criterion

spar: A list with
  • all: All smoothing parameters searched
  • final: The smoothing parameter used
  • final.idx: The index of the final spar

enp: A list with
  • raw: Raw effective number of parameters
  • logistic: Effective number of parameters after fitting logistic curve as a correction
  • final: The effective number of parameters in the final model
  • good.idx: The index of the selected effective number of parameters

fit: A list with
  • intercept: The fitted intercept
  • covariate.idx: The index of covariates
  • f.covariate: Each additive smooth function evaluated at the covariates
  • f: Fitted smoothing function
  • beta: Estimated regression coefficients
  • H: Expanded design matrix
  • asym.vcov: Asymptotic variance-covariance matrix for estimated parameters

NPLL: A list with
  • NPLL: Negative penalized log likelihood
  • logLik: Log likelihood
  • penalty: Penalty term
  • saturated.ll: Saturated log likelihood
**Note**

When spar is too small, the results need to be treated cautiously. It is advisable to plot the results as a check.

**Author(s)**

Long Qu <rtistician@gmail.com>

**References**


**See Also**

`plot.hisemit`, `fitted.hisemit`, `coef.hisemit`, `vcov.hisemit`, `residuals.hisemit`, `logLik.hisemit`, `confint.hisemit`, `plot.hisemit`, `hisemi-package`, `pi0-package`

**Examples**

# See the examples for the hisemi-package.

---

**plot.hisemit**

*Plot a hisemit object*

**Description**

Plot an object of class hisemit

**Usage**

```r
## S3 method for class 'hisemit'
plot(x, type = c("tuning", "residual"), ...)  
plotHisemitResid(obj, y.type = c("hist", "scatter"), x.type = c("lfdr", "pi0", "f"), ...)  
plotHisemitTuning(obj, SE = FALSE, add = FALSE, ...)  
```

**Arguments**

- `x, obj` The object of class hisemit
- `type` Either 'tuning' or 'residual', the type of plot requested.
- `...` See Details.
- `y.type` Either 'hist' or 'scatter', the type of residual plot requested.
- `x.type` One of 'lfdr', 'pi0' or 'f', the x-axis when y.type='scatter'.
SE

A logical scalar, indicating whether standard error bars should be added to the plot.

add

A logical scalar, indicating whether a new plot should be generated or adding lines to the existing plot.

Details

The generic function plot.hisemit calls either plotHisemitResid or plotHisemitTuning depending on type. For plot.hisemit, the ... is the additional arguments to be passed to plotHisemitResid or plotHisemitTuning. For residual plot (plotHisemitResid), the ... is the additional arguments to be passed to hist when y.type='hist'; and to residuals.hisemit when y.type='scatter'. For tuning plot (plotHisemitTuning), the ... is the additional arguments to be passed to either plot or lines depending on add.

Value

For histograms of residuals, an object from hist is returned. For scatter plot of residuals, an object from plot is returned. For tuning plot, a matrix with the range of tuning criterion is returned.

Author(s)

Long Qu <rtistician@gmail.com>

References


See Also

plot.hisemit, fitted.hisemit, coef.hisemit, vcov.hisemit, residuals.hisemit, logLik.hisemit, confint.hisemit, print.hisemit, penLik.EMNewton

Description

Print summaries of a hisemit object.

Usage

## S3 method for class 'hisemit'
print(x, ...)
## S3 method for class 'hisemit'
summary(object, ...)
residuals.hisemit

Arguments

- **object**: The hisemit object to be printed.
- **residual.type**: A character scalar specifying the type of residuals to be extracted. Currently only 'deviance' is supported.
- **...**: Additional arguments to be passed to `print.default` or `summary.default` for the list object.

Details

Currently, the function directly calls the corresponding method for the list object.

Value

The same as the results from the corresponding method for the list object.

Author(s)

Long Qu <rtistician@gmail.com>

References


See Also

- `plot.hisemit`
- `fitted.hisemit`
- `coef.hisemit`
- `vcov.hisemit`
- `residuals.hisemit`
- `logLik.hisemit`
- `confint.hisemit`
- `print.hisemit`
- `penLik.EMNewton`

---

**residuals.hisemit**: Extract residuals from a hisemit object

Description

Extract residuals from a hisemit object

Usage

```r
## S3 method for class 'hisemit'
residuals(object, residual.type = "deviance", ...)
```

Arguments

- **object**: A hisemit object
- **residual.type**: A character scalar specifying the type of residuals to be extracted. Currently only 'deviance' is supported.
- **...**: Currently not used.
Value

A numeric vector residuals.

Author(s)

Long Qu <rtistician@gmail.com>

References


See Also

plot.hisemit, fitted.hisemit, coef.hisemit, vcov.hisemit, residuals.hisemit, loglik.hisemit, confint.hisemit, print.hisemit, penLik.EMNewton


t  Pseudo-random number generation from t-distribution

Description

This is a slightly modified implementation of stats:rt.

Usage

t(n, df, ncp)

Arguments

n, df  Exactly as in stats:rt.
ncp  A numeric vector of non-centrality parameters.

Details

This function corrects the bug in stats:rt that causes trouble when ncp is a vector. See Bug 17306.

Value

A numeric vector of length n which are pseudo-random numbers from the requested t-distribution.

Author(s)

Long Qu modified the code of stats:rt.
See Also

stats::rt

Examples

set.seed(99927220)
rt(5, 1) # central t
rt(5, 1, 1:5*5) # noncentral t where stats::rt throws warnings
## Not run:
rt(5, 1, c(NA, 1:4*5)) # noncentral t where stats::rt returns all NaN's
rt(5, 1, c(1:4*5, NA)) # noncentral t where stats::rt throws extra warnings
## End(Not run)

Description

This function fits a null model to t-statistics, i.e., a two-component mixture, with one component
being central t-distribution, the other component being scaled central t-distribution with scale pa-
parameter larger than 1. The mixing proportion for the central t-distribution is pi0.

Usage

scaledTMix.null(tstat, df, starts =
  list(pi0 = seq(0.1, 0.99, length = 20),
       scale = 2*seq(0.01, log2(max(abs(tstat)))),
       length = 20))

Arguments

tstat A numeric vector of t-statistics.
df A numeric scalar or vector of the same length as tstat, giving the degrees of
    freedom for the tstat.
starts A list of two components, pi0 and scale. Each being a numeric vector, which
    defines the grid for searching starting values.

Details

The function finds maximum likelihood estimates of pi0 and scale. pi0 should lie between 0 and
1. scale should be larger than 1. The L-BFGS-B method is used in optimization function optim.
The parncp function with zeromean=TRUE fits the same model, except that it returns ncpest object
instead of hisemit object.
Value
A hesimit object.

Author(s)
Long Qu <rtistician@gmail.com>

References

See Also
parncp, scaledTMix.sat, scaledTMix.psat

Examples
set.seed(99927220)
(tstat=rt(5,1))
scaledTMix.null(tstat,1)

scaledTMix.psat       Fits a partially saturated model to t-statistics

Description
Fits two-component mixture model to t-statistics, where each t-statistic has a different mixing proportion pi0, but all t-statistics shares a common scale factor.

Usage
scaledTMix.psat(tstat, df, upper0 = 2)

Arguments
tstat        A numeric vector t-statistics
df          A numeric scalar or vector of degrees of freedom
upper0       A numeric scalar giving the initial upper bound to search for scale factor. It will be automatically increased if the initial bound is not appropriate.

Details
This function fits a two-component mixture model, with a central t component with probability pi0, and a scaled central t component with scale factor greater than 1. The model assumes a different pi0 for each t-statistic, but a common scale factor for all t-statistics. Maximum likelihood estimates are obtained. upper0 is only used as a hint of the upper bound of the scale factor. If it is too small, it will be automatically increased.
Value

A numeric scalar, being estimated scale factor, with attributes

- **equiv.sd.ncp**: A numeric scalar being the equivalent standard deviation of the noncentrality parameters, given the noncentrality parameter being nonzero.
- **df**: degrees of freedom
- **fit**: results from `optimize`
- **n2ll**: Negative 2 times the log likelihood
- **pi0**: A numeric vector of 0 or 1, which are the maximum likelihood estimate of pi0

Author(s)

Long Qu <rtistician@gmail.com>

See Also

- `scaledTMix.null`, `scaledTMix.sat`

Examples

```r
cat(1)
scaledTMix.psat(tstat)
```

Description

Fit saturated model to t-statistics, i.e., a two-component mixture model (a central t and a scaled central t with scale greater than 1) to each t-statistics separately.

Usage

`scaledTMix.sat(tstat, df)`

Arguments

- **tstat**: A numeric vector of t-statistics.
- **df**: A numeric scalar or vector of the same length as `tstat`, giving the degrees of freedom for the `tstat`.

Details

This function assumes each t-statistics coming from either a central t-distribution or a scaled central t-distribution. Each t-statistic has a different mixing proportion pi0, whose maximum likelihood estimate will be either 0 or 1. Each t-statistic has a different scale parameter. If pi0=1, the scale parameter will be 1; if pi0=0, the scale parameter will be greater than 1.
Value

A numeric vector of estimated scale parameters, with two attributes

pi0 A numeric vector of estimated pi0
logLik A numeric vector of log likelihood

Note

Whenever the absolute value of the tstat is less than 1, pi0 will be estimated to be 1 and the scale will also be 1. Otherwise, the pi0 will be estimated to be 0 and scale will be the absolute value of tstat.

Author(s)

Long Qu <rtistician@gmail.com>

See Also

scaledTMix.null, scaledTMix.psat

Examples

set.seed(99927220)
(tstat=rt(5,1))
scaledTMix.sat(tstat,1)
Arguments

tstat A numeric vector t-statistics
x A numeric matrix of covariates, with nrow(x) being length(tstat)
df A numeric scalar or vector of degrees of freedom
starts An optional numeric vector of starting values. The first element is the r, i.e. log(scale-1). The second parameter is the intercept. The remaining elements are the starting values for the B-spline coefficients (removing the first basis) for each x. When this argument is not provided, the code starts with a global constant model that is easiest to fit, and then increase the order gradually using the warm starts from lower order fits.
pen.order A numeric scalar of the order of derivatives of which squared integration will be used as roughness penalty. Note: The final order of the global polynomial is always pen.order-1.
optim.method A character scalar specifying the method of optimization.
newton.iter.max A numeric scalar specifying the maximum number of iterations in Newton method.
scale.conv A small numeric scalar specifying the convergence criterion for the scale parameter.
lfdr.conv A small numeric scalar specifying the convergence criterion for the local false discovery rates.
NPLL.conv A small numeric scalar specifying the convergence criterion for the negative penalized log likelihood.
debugging A logical scalar. If TRUE, then dump.frame will be called whenever error occurs.
plotit A logical scalar specifying whether a plot should be generated.

Value

An list of class hisemit:
lfdr: A numeric vector of local false discovery rates.
model A list of tstat, df and x, which are the same as arguments
scale.fact: A list with
  • scale.fact: Scale factor
  • sd.ncp: Equivalent standard deviation of noncentrality parameters
  • r: A reparameterization of scale.fact
  • t.cross: $\sqrt{t(df \times (s^2/(2/(df+1)))^{-1}/(1-s^2(-2\times df/(df+1)))}$
where s is the scale.fact
pi0: A numeric vector of mixing proportions for the central t component
tuning: A list with
  • mean: Mean criterion
  • var: Variance of criterion across observations
tPoly.newton

- grp: Cross-validation group membership
- method: The tuning method used.
- final: The minimum mean criterion

spar: A list with
- all: All smoothing parameters searched
- final: The smoothing parameter used
- final.idx: The index of the final spar

enp: A list with
- raw: Raw effective number of parameters
- logistic: Effective number of parameters after fitting logistic curve as a correction
- final: The effective number of parameters in the final model
- good.idx: The index of the selected effective number of parameters

fit: A list with
- intercept: The fitted intercept
- covariate.idx: The index of covariates
- f.covariate: Each additive smooth function evaluated at the covariates
- f: Fitted smoothing function
- beta: Estimated regression coefficients
- H: Expanded design matrix
- asym.vcov: Asymptotic variance-covariance matrix for estimated parameters

NPLL: A list with
- NPLL: Negative penalized log likelihood
- logLik: Log likelihood
- penalty: Penalty term
- saturated.ll: Saturated log likelihood

Author(s)
Long Qu <rtistician@gmail.com>

References

See Also
penLik, EMNewton, plot.hisemit, fitted.hisemit, coef.hisemit, vcov.hisemit, residuals.hisemit, logLik.hisemit, confint.hisemit, plot.hisemit, hisemi-package, pi0-package

Examples

# See the example for the package.
vcov.hisemit

Extract the asymptotic variance-covariance matrix of a hisemit object

Description

Extract the asymptotic variance-covariance matrix of a hisemit object.

Usage

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

Arguments

- `object`: A hisemit object.
- `...`: Currently not used.

Details

Variance-covariance matrix for the fitted parameters.

Value

A numerical matrix.

Author(s)

Long Qu <ristician@gmail.com>

References


See Also

- `plot.hisemit`
- `fitted.hisemit`
- `coef.hisemit`
- `vcov.hisemit`
- `residuals.hisemit`
- `logLik.hisemit`
- `confint.hisemit`
- `print.hisemit`
- `penLik.EMNewton`
Index

*Topic arith
  directSum, 6
*Topic array
  directSum, 6
*Topic distribution
  rt, 21
*Topic hplot
  plot.hisemit, 18
*Topic math
  directSum, 6
*Topic methods
  print.hisemit, 19
*Topic models
  coef.hisemit, 4
  confint.hisemit, 5
  fitted.hisemit, 8
  hisemi-package, 2
  logistic.enp, 9
  logit, 10
  logLik.hisemit, 11
  penLik.EMNewton, 15
  residuals.hisemit, 20
  scaledTMix.null, 22
  scaledTMix.psat, 23
  scaledTMix.sat, 24
  tPoly.newton, 25
  vcov.hisemit, 28
*Topic nonlinear
  logistic.enp, 9
*Topic optimize
  EMupdate, 7
  hisemi-package, 2
  NRupdate, 13
  penLik.EMNewton, 15
  scaledTMix.null, 22
  scaledTMix.psat, 23
  scaledTMix.sat, 24
  tPoly.newton, 25
*Topic package
  hisemi-package, 2
*Topic print
  print.hisemit, 19
*Topic regression
  coef.hisemit, 4
  fitted.hisemit, 8
  hisemi-package, 2
  plot.hisemit, 18
  residuals.hisemit, 20
  vcov.hisemit, 28
*Topic smooth
  n.knots, 12
  OsplinePen, 14
  bsplinepen, 14
  coef.hisemit, 2, 4, 6, 9, 11, 18–21, 27, 28
  confint.hisemit, 2, 4, 5, 6, 9, 11, 18–21, 27, 28
directSum, 2, 6
EMupdate, 2, 7, 14
fitted.hisemit, 2, 4, 6, 8, 9, 11, 18–21, 27, 28
hisemi (hisemi-package), 2
hisemi-package, 2
hist, 19
lines, 19
logistic.enp, 9
logit, 10
logLik.hisemit, 2, 4, 6, 9, 11, 18–21, 27, 28
make.link, 10, 11
n.knots, 12
nearPD, 13
nls, 10
NRupdate, 2, 8, 13

optim, 22
optimize, 24
OsplinePen, 14

parncp, 22, 23
penLik.EMNewton, 2, 4, 6–9, 11, 13, 14, 15,
  19–21, 27, 28
plot, 19
plot.hisemit, 2, 4, 6, 9, 11, 18, 18, 19–21,
  27, 28
plotHisemitResid(plot.hisemit), 18
plotHisemitTuning(plot.hisemit), 18
print.default, 20
print.hisemit, 2, 4, 6, 9, 11, 19, 19, 20, 21,
  28
residuals.hisemit, 2, 4, 6, 9, 11, 18–20, 20,
  21, 27, 28
rt, 21

scaledTMix.null, 3, 22, 24, 25
scaledTMix.psat, 3, 23, 23, 25
scaledTMix.sat, 3, 23, 24, 24
stats:rt, 21, 22
summary.default, 20
summary.hisemit(print.hisemit), 19
tPoly.newton, 25

vcov.hisemit, 3, 4, 6, 9, 11, 18–21, 27, 28, 28