Package 'DALSM'

January 20, 2025

Type Package

Title Nonparametric Double Additive Location-Scale Model (DALSM)

Version 0.9.1

Maintainer Philippe Lambert <p.lambert@uliege.be>

BugReports https://github.com/plambertULiege/DALSM/issues

Description
Fit of a double additive location-scale model with a nonparametric error distribution from possibly right- or interval censored data. The additive terms in the location and dispersion submodels, as well as the unknown error distribution in the location-scale model, are estimated using Laplace P-splines. For more details, see Lambert (2021) <doi:10.1016/j.csda.2021.107250>.

License GPL-3

URL <https://github.com/plambertULiege/DALSM>

Encoding UTF-8

LazyData true

Imports cubicBsplines, MASS, plyr

RoxygenNote 7.2.3

NeedsCompilation no

Author Philippe Lambert [aut, cre] (<https://orcid.org/0000-0002-3670-3328>)

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-10-02 16:50:02 UTC

Contents

centeredBasis.gen 2
DALSM 3
DALSM.object 5
DALSM_additive 8
DALSM_IncomeData 10
Dens1d 11

Dens1d.object 12
densityLPS 14
densLPS.object 16
logDet.fun 18
Pcal.fun 18
plot.DALSM 19
plot.densLPS 21
plotRegion 23
predict.DALSM 24
print.DALSM 25
print.densLPS 26
qknots 27

Index **29**

`centeredBasis.gen` *Generation of a recentered cubic B-spline basis matrix in additive models*

Description

Generation of a cubic B-spline basis matrix with recentered columns to handle the identifiability constraint in additive models. See Wood (CRC Press 2017, pp. 175-176) for more details.

Usage

```
centeredBasis.gen(x, knots, cm = NULL, pen.order = 2)
```

Arguments

`x`	vector of values where to compute the "recentered" B-spline basis.
`knots`	vector of knots (that should cover the values in <x>).
`cm`	(Optional) values subtracted from each column of the original B-spline matrix.
`pen.order`	penalty order for the B-spline parameters (Default: 2).

Value

List containing

- `B` : centered cubic B-spline matrix (with columns recentered to have mean 0 over equi-spaced x values on the range of the knots).
- `Dd` : difference matrix (of order <pen.order>) for the associated centered B-spline matrix.
- `Pd` : penalty matrix (of order <pen.order>) for the associated centered B-spline matrix.
- `K` : number of centered B-splines in the basis.
- `cm` : values subtracted from each column of the original B-spline matrix. By default, this is a vector containing the mean of each column in the original B-spline matrix.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

Examples

```
x = seq(0,1,by=.01)
knots = seq(0,1,length=5)
obj = centeredBasis.gen(x,knots)
matplot(x,obj$B,type="l",ylab="Centered B-splines")
colMeans(obj$B)
```

DALSM *Fit a double additive location-scale model (DALSM) with a flexible error distribution*

Description

Fit a location-scale regression model with a flexible error distribution and additive terms in location (=mean) and dispersion (= log(sd)) using Laplace P-splines from potentially right- and interval-censored response data.

Usage

```
DALSM(y, formula1,formula2, data,
       K1=10, K2=10, pen.order1=2, pen.order2=2,
       b.tau=1e-4, lambda1.min=1, lambda2.min=1,
       phi.0=NULL,psi1.0=NULL,psi2.0=NULL,lambda1.0=NULL,lambda2.0=NULL,
       REML=FALSE, diag.only=TRUE, Normality=FALSE, sandwich=TRUE,
       K.error=20, rmin=NULL,rmax=NULL,
       ci.level=.95, iterlim=50,verbose=FALSE)
```

Arguments

y	n-vector of responses or (nx2)-matrix/data.frame when interval-censoring (with the 2 elements giving the interval bounds) or right-censoring (when the element in the 2nd column equals Inf).
formula1	model formula for location (i.e. for the conditional mean).
formula2	model formula for dispersion (i.e. for the log of the conditional standard deviation).
data	data frame containing the model covariates.

K1	(optional) number of B-splines to describe a given additive term in the location submodel (default: 10).
K2	(optional) number of B-splines to describe a given additive term in the dispersion submodel (default: 10).
pen.order1	(optional) penalty order for the additive terms in the location submodel (default: 2).
pen.order2	(optional) penalty order for the additive terms in the dispersion submodel (default: 2).
b.tau	(optional) prior on penalty parameter τ is Gamma(1,b.tau=1e-4) (for additive terms) (default: 1e-4).
lambda1.min	(optional) minimal value for the penalty parameters in the additive model for location (default: 1.0).
lambda2.min	(optional) minimal value for penalty parameters in the additive model for log-dispersion (default: 1.0).
phi.0	(optional) initial values for the spline parameters in the log-hazard of the standardized error distribution.
psi1.0	(optional) initial values for the location submodel parameters.
psi2.0	(optional) initial values for the dispersion submodel parameters.
lambda1.0	(optional) initial value for the J1 penalty parameters of the additive terms in the location submodel.
lambda2.0	(optional) initial value for the J2 penalty parameters of the additive terms in the dispersion submodel.
REML	(optional) logical indicating if a REML correction is desired to estimate the dispersion parameters (default: FALSE).
diag.only	(optional) logical indicating if only the diagonal of the Hessian needs to be corrected during REML (default: TRUE).
Normality	(optional) logical indicating if Normality is assumed for the error term (default: FALSE).
sandwich	(optional) logical indicating if sandwich variance estimators are needed for the regression parameters for a NP error distribution (when Normality=FALSE) ; it is forced to be TRUE when Normality=TRUE.
K.error	(optional) number of B-splines to approximate the log of the error hazard on (rmin,rmax) (default: 20).
rmin	(optional) minimum value for the support of the standardized error distribution (default: min(y)-sd(y)).
rmax	(optional) maximum value for the support of the standardized error distribution (default: max(y)+sd(y)).
ci.level	(optional) nominal level for the reported credible intervals (default: .95).
iterlim	(optional) maximum number of iterations (after which the algorithm is interrupted with a non-convergence diagnostic) (default: 50).
verbose	(optional) logical indicating whether estimation step details should be displayed (default: FALSE).

Value

An object of class DASLM containing several components from the fit, see DALSM.object for details. A summary can be printed using print.DALSM or plotted using plot.DALSM.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

DALSM.object, print.DALSM, plot.DALSM.

Examples

```
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
print(fit)
plot(fit)
```

DALSM.object — *Object resulting from the fit of a double additive location-scale model (DALSM).*

Description

An object returned by the DALSM function: this is a list with various components related to the fit of a double additive location-scale model using Laplace P-splines.

Value

A DALSM object has the following elements:

Essential part:

- converged : logical convergence indicator.
- derr : estimated standardized error distribution returned as a densLPS.object.

- psi1 : estimated regression parameters for location (fixed effects, B-spline coefs for the J1 additive terms).
- psi2 : estimated regression parameters for dispersion (fixed effects, B-spline coefs for the J2 additive terms).
- fixed.loc : matrix with the estimated fixed effects (est,se,ci.low,ci.up) in the location sub-model.
- fixed.disp : matrix with the estimated fixed effects (est,se,ci.low,ci.up) in the dispersion sub-model.
- mu : n-vector with the fitted conditional mean.
- sd : n-vector with the fitted conditional standard deviation.

Additional elements:

- data : the original data frame used when calling the DALSM function.
- phi : estimated B-spline coefs for the log-hazard of the error distribution.
- K.error : number of B-splines used to approximate the log of the error hazard.
- rmin, rmax : minimum and maximum values for the support of the standardized error distribution.
- knots.error : equidistant knots on (rmin,rmax) used to specify the B-spline basis for the log of the error hazard.
- bread.psi1, Sand.psi1, Cov.psi1: estimated Variance-Covariance matrix for ψ_1.
- U.psi1 : gradient for ψ_1.
- bread.psi2, Sand.psi2, Cov.psi2: estimated Variance-Covariance matrix for ψ_2.
- U.psi2 : gradient for ψ_2.
- U.psi : gradient for $\psi = (\psi_1, \psi_2)$.
- Cov.psi : variance-covariance for $\psi = (\psi_1, \psi_2)$.
- regr1 : object generated by DesignFormula for the specified submodel for location.
- regr2 : object generated by DesignFormula for the specified submodel for dispersion.
- res : n-vector or nx2 matrix (if IC data) with the standardized residuals for the fitted model.
- expctd.res : n-vector with observed standardized residual for a non RC unit, or their expected value if right-censored.
- REML : logical indicating whether REML estimation was performed.
- n : the sample size.
- n.uncensored : number of non-censored response data.
- event : n-vector of event indicators (1: non right-censored ; 0: right censoring).
- is.IC : n-vector with interval censoring indicators.
- n.IC : number of interval-censored response data.
- n.RC : number of right-censored response data.
- perc.obs : percentage of exactly observed response data.
- perc.IC : percentage of interval-censored response data.

- `perc.RC` : percentage of right-censored response data.
- `cred.int` : nominal level for the reported credible intervals.
- `alpha` : user-specified α with Bayesian $(1-\alpha)$ credible intervals reported.
- `sandwich` : logical indicating if variance-covariance and standard errors computed using sandwich estimator in the NP case.
- `diag.only` : logical indicating if the correction to the Hessian under REML only concerns diagonal elements.
- `iter` : number of iterations.
- `elapsed.time` : time required by the model fitting procedure.

If there are additive terms in the location submodel:

- `K1` : number of B-splines used to describe an additive term in the location submodel.
- `xi1` : matrix with the selected log penalty parameters for the J1 additive terms in the location submodel (point estimate, se, ci.low, ci.up.
- `U.xi1` : gradient for the log of the penalty parameters for the J1 additive terms in the location submodel.
- `U.lambda1` : gradient for the penalty parameters for the J1 additive terms in the location submodel.
- `Cov.xi1` : estimated Variance-Covariance matrix for the parameters involved in the J1 additive terms in the location submodel.
- `lambda1.min` : minimal value for the penalty parameters in the additive submodel for location.
- `lambda1` : matrix with the selected penalty parameters for the J1 additive terms in the location submodel (point estimate, se, ci.low, ci.up).
- `ED1` : matrix with the effective dimensions for each of the J1 additive terms in the location submodel (point estimate,ci.low,ci.up).

If there are additive terms in the dispersion submodel:

- `K2` : number of B-splines used to describe an additive term in the dispersion submodel.
- `xi2` : matrix with the selected log penalty parameters for the J2 additive terms in the dispersion submodel (point estimate, se, ci.low, ci.up).
- `U.xi2` : gradient for the log of the penalty parameters for the J2 additive terms in the dispersion submodel.
- `U.lambda2` : gradient for the penalty parameters for the J2 additive terms in the dispersion submodel.
- `Cov.xi2` : estimated Variance-Covariance matrix for the parameters involved in the J2 additive terms in the dispersion submodel.
- `lambda2.min` : minimal value for the penalty parameters in the additive submodel for dispersion.
- `lambda2` : matrix with the selected penalty parameters for the J2 additive terms in the dispersion submodel (point estimate, se, ci.low, ci.up).
- `ED2` : matrix with the effective dimensions for each of the J2 additive terms in the dispersion submodel (point estimate,ci.low,ci.up).

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

DALSM, print.DALSM, plot.DALSM, densityLPS, densLPS.object

DALSM_additive *Extraction of the estimated additive terms in a* DALSM.object

---

Description

Extract the estimated additive terms with their standard errors from a DALSM.object resulting from the fit of a DALSM model. In addition to the estimated functions in the location and dispersion sub-models, their values on a regular grid covering the observed covariate values are reported together with credible intervals. The mean effective coverage of these pointwise credible intervals for the additive terms with respect to given (optional) reference functions (such as the ones for the 'true' additive terms used to generate data in a simulation study) can also be computed.

Usage

```
DALSM_additive(obj.DALSM, ngrid=101,
       true.loc=NULL, true.disp=NULL, ci.level=NULL,
       verbose=FALSE)
```

Arguments

obj.DALSM	a DALSM.object
ngrid	(optional) grid size of covariate values where the additive terms are calculated (default: 101).
true.loc	(optional) list of functions containing the 'true' additive terms in the location sub-model.
true.disp	(optional) list of functions containing the 'true' additive terms in the dispersion sub-model.
ci.level	(optional) level of credible intervals.
verbose	logical indicating whether the computed corverages should be printed out (default: TRUE).

Value

It returns an invisible list containing:

- `J1` : number of additive terms in the location sub-model.
- `labels.loc` : labels of the additive terms in the location sub-model.
- `f.loc.grid` : list of length `J1` with, for each additive term, a list of length 3 with elements 'x': a vector of `ngrid` values for the covariate ; 'y.mat': a matrix with 3 columns (est,low,up) giving the additive term and its pointwise credible region ; se: the standard error of the additive term on the x-grid.
- `f.loc` : a list of length `J1` with, for each additive term <x>, a list with f.loc$x: a function computing the additive term f.loc(x) for a given covariate value 'x' ; attributes(f.loc$x): support, label, range.
- `se.loc` : a list of length `J1` with, for each additive term <x>, a list with se.loc$x: a function computing the s.e. of f(x) for a given covariate value 'x' ; attributes(se.loc$x): support, label, range.
- `coverage.loc` : if `true.loc` is provided: a vector of length `J1` giving the average effective coverage of pointwise credible intervals for each of the additive terms in the location sub-model.
- `J2` : number of additive terms in the dispersion sub-model.
- `labels.disp` : labels of the additive terms in the dispersion sub-model.
- `f.disp.grid` : list of length `J2` with, for each additive term, a list of length 3 with elements 'x': a vector of `ngrid` values for the covariate ; 'y.mat': a matrix with 3 columns (est,low,up) giving the additive term and its pointwise credible region ; se: the standard error of the additive term on the x-grid.
- `f.disp` : a list of length `J2` with, for each additive term <x>, a list with f.disp$x: a function computing the additive term f.disp(x) for a given covariate value 'x' ; attributes(f.disp$x): support, label, range.
- `se.disp` : a list of length `J2` with, for each additive term <x>, a list with se.disp$x: a function computing the s.e. of f(x) for a given covariate value 'x' ; attributes(se.disp$x): support, label, range.
- `coverage.disp` : if <true.disp> is provided: a vector of length `J2` giving the average effective coverage of pointwise credible intervals for each of the additive terms in the dispersion sub-model.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

`DALSM.object`, `DALSM`, `print.DALSM`, `plot.DALSM`.

Examples

```
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
obj = DALSM_additive(fit)
str(obj)

## Visualize the estimated additive terms for Age
## ... in the location submodel
with(obj$f.loc.grid$age, matplot(x,y.mat,
                                 xlab="Age",ylab="f.loc(Age)",
                                 type="l",col=1,lty=c(1,2,2)))
## ... and in the dispersion submodel
with(obj$f.disp.grid$age, matplot(x,y.mat,
                                 xlab="Age",ylab="f.disp(Age)",
                                 type="l",col=1,lty=c(1,2,2)))
## Also report their values for selected age values
obj$f.loc$age(c(30,40,50)) ; obj$f.disp$age(c(30,40,50))
## ... together with their standard errors
obj$se.loc$age(c(30,40,50)) ; obj$se.disp$age(c(30,40,50))
```

DALSM_IncomeData *Income data*

Description

Income (in euros) available per person in 756 Belgian households (ESS 2016 ; Lambert 2021, Section 4) for respondents aged 25-55 when the main source of income comes from wages or salaries. The data were extracted from the European Social Survey (2016) and reworked to quantify the resources available at the individual level in the selected Belgian households.

Usage

```
data(DALSM_IncomeData)
```

Format

A data frame with 756 rows and 5 columns:

- `inc.low` : lower bound for the individual income data.
- `inc.up` : upper bound for the individual income data (`Inf` if right-censored).
- `twoincomes` : `0`: one-income household ; `1`: two-income household.
- `age` : age of the respondent.
- `eduyrs` : number of years of education completed.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

European Social Survey Round 8 Data (2016) Data file edition 2.1. NSD - Norwegian Centre for Research Data, Norway.

`Dens1d` *Object creation for density estimation from right- or interval-censored data*

Description

Object creation for density estimation from right- or interval-censored data using function densityLPS.

Usage

```
Dens1d(y, event=NULL, ymin=NULL, ymax=NULL,
       K=25, equid.knots=TRUE, pen.order=2, nbins=501)
```

Arguments

`y`	a n-vector (if no interval-censored data) or a nx2 matrix (left and right limits of the interval for IC data ; right limit set to Inf for right-censored data).
`event`	a n-vector of observation indicators (0: right-censored ; 1: exactly observed or interval-censored).
`ymin`	left limit of the variable support.
`ymax`	right limit of the variable support.
`K`	number of B-splines in the basis to approximate the log-hazard.
`equid.knots`	logical indicating if equidistants knots are desired.
`pen.order`	penalty order when equidistant knots (otherwise: penalty matrix computed to penalize the second derivative).
`nbins`	number of small bins used for quadrature and approximations.

Value

A Dens1d.object, i.e. a list with summary measures and precomputed components required for density estimation using `densityLPS`.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

Dens1d.object, densityLPS.

Examples

```
library(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
head(resp,n=20)
temp = Dens1d(y=resp,ymin=0) ## Create Dens1d object from positive censored data
obj = densityLPS(temp) ## Density estimation from IC & RC data
plot(obj) ## Visualize the estimated density
```

Dens1d.object — *Object created by* Dens1d *to prepare for density estimation from censored data using* densityLPS

Description

An object returned by function Dens1d to prepare for density estimation with given mean and variance from censored data using function densityLPS.

Value

A Dens1d.object is a list with the following elements:

- n : total sample size.
- y : a n-vector if no interval-censored data, a nx2 matrix otherwise (for interval-censored data: left and right limits of the interval ; for right-censored data: the right limit is set to Inf).
- event : a n-vector of observation indicators (0: right-censored ; 1: exactly observed or interval-censored).
- ymin : left limit of the variable support.
- ymax : right limit of the variable support.
- is.uncensored : boolean n-vector indicating if the corresponding <y> value is not right- or interval-censored.

- `n.uncensored` : total number of uncensored data.
- `is.IC` : boolean n-vector indicating if the corresponding `y` value is interval-censored.
- `n.IC` : total number of interval-censored data.
- `is.RC` : n-vector of booleans indicating if the corresponding `y` value is right-censored.
- `n.RC` : total number of right-censored data.
- `ylow, yup` : n-vector with the lower and upper limits of the interval data (when interval-censored). When `y` is exactly observed or right-censored, the two values are identical.
- `ymid` : n-vector containing the mean of `y.low` and `y.up`.
- `K` : number of B-splines in the basis used to model the log-hazard.
- `knots` : vector of knots for the B-spline basis.
- `pen.order` : penalty order.
- `Pd` : penalty matrix in the P-spline model.
- `nbins` : number of small bins used to partition the variable support.
- `bins` : small bins of equal width used to partition the variable support (cf. binning).
- `ugrid` : midpoints of the small bins.
- `dbins` : width of a small bin.
- `Bbins` : ((nbins +1) x K)-matrix containing the B-spline basis evaluated at the bin limits.
- `C` : (n x nbins) matrix of event or censoring indicators C_{ij} for unit 'i' and bin 'j'. For a unit with IC data, the bins with an non-empty intersection with the interval are indicated. When the unit is associated to a precise event time or censoring time in bin 'j', then $C_{ij} = 1$ and 0 for other bins.
- `Bgrid` : (nbins x K)-matrix containing the B-spline basis evaluated at the bin midpoints.
- `fgrid` : nbins-vector with the estimated density values at the bin midpoints.
- `rgrid` : nbins-vector with the number of units 'still at risk' of an "event" in a given bin.
- `BsB, ev` : technical quantities used in the estimation procedure, see the code in Dens1d.R for more details.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

`Dens1d`, `densityLPS`, `densLPS.object`

`densityLPS` *Constrained density estimation from censored data with given mean and variance using Laplace P-splines*

Description

P-spline estimation of the density (pdf), cumulative distribution (cdf), hazard and cumulative hazard functions from interval- or right-censored data under possible marginal mean and/or variance constraints. The penalty parameter τ tuning the smoothness of the log-hazard can be selected using the Laplace P-splines (LPS) method maximizing an approximation to the marginal posterior of τ (also named the 'evidence') or using Schall's method.

Usage

```
densityLPS(obj.data,
       is.density=TRUE, Mean0=NULL, Var0=NULL,
       fixed.penalty=FALSE, method=c("LPS","Schall"),
       fixed.phi=FALSE,phi.ref=NULL, phi0=NULL,tau0=exp(5),tau.min=.1,
       verbose=FALSE)
```

Arguments

`obj.data`	a list created from potentially right- or interval-censored data using `Dens1d`. It includes summary statistics, the assumed density support, the knots for the B-spline basis, etc.
`is.density`	(optional) logical indicating whether the estimated density should integrate to 1.0 over the range of the knots in obj.data$knots (default: TRUE).
`Mean0`	(optional) constrained value for the mean of the fitted density (defaut: NULL).
`Var0`	(optional) constrained value for the variance of the fitted density (defaut: NULL).
`fixed.penalty`	(optional) logical indicating whether the penalty parameter should be selected from the data (`fixed.penalty=FALSE`) or fixed (`fixed.penalty=TRUE`) at its initial value τ_0.
`method`	method used for the selection of the penalty parameter: "LPS" (by maximizing the marginal posterior for τ, cf. "Laplace P-splines") or "Schall" (Schall's method).
`fixed.phi`	(optional) logical indicating whether the spline parameters are fixed (`fixed.phi=TRUE`) or estimated from the data (default: `fixed.phi=FALSE`).
`phi.ref`	(optional) reference value for the spline parameters with respect to which deviations are penalized (default: zero vector).
`phi0`	starting value for the spline parameters (default: spline parameters corresponding to a Student density with 5 DF).
`tau0`	(optional) initial value for the penalty parameter τ (default: exp(5)).
`tau.min`	(optional) minimal value for the penalty parameter τ (default: .1).
`verbose`	(optional) logical indicating whether estimation step details should be displayed (default: FALSE).

Value

a densLPS.object containing the density estimation results.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

densLPS.object, print.densLPS, plot.densLPS, Dens1d.object, Dens1d.

Examples

```
library(DALSM)

## Example 1: estimation of the error density in a DALSM model
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
plot(fit$derr)  ## Plot the estimated error density
print(fit$derr) ## ... and provide summary statistics for it

## Example 2: density estimation from censored income data
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
head(resp,n=20)
temp = Dens1d(y=resp,ymin=0) ## Create Dens1d object from positive censored data
obj = densityLPS(temp) ## Density estimation from IC & RC data
print(obj) ## Summary information on the estimated density
plot(obj,hist=TRUE) ## Visualize the estimated density
legend("topright",col=c("black","grey"),lwd=c(2,20),
       legend=c("Fitted density","Pseudo-data"),bty="n")

## Example 3: density estimation from simulated RC and IC data
## Data generation
set.seed(123)
n = 500 ## Sample size
x = rgamma(n,10,2) ## Exact (unobserved) data
width = runif(n,1,3) ## Width of the IC data (mean width = 2)
w = runif(n) ## Positioning of the exact data within the interval
xmat = cbind(pmax(0,x-w*width),x+(1-w)*width) ## Generated IC data
t.cens = rexp(n,1/15) ## Right-censoring values
```

```
idx.RC = (1:n)[t.cens<x] ## Id's of the right-censored units
xmat[idx.RC,] = cbind(t.cens[idx.RC],Inf) ## Data for RC units: (t.cens,Inf)
head(xmat,15)
## Density estimation with mean and variance constraints
obj.data = Dens1d(xmat,ymin=0) ## Prepare the data for estimation
obj = densityLPS(obj.data,Mean0=10/2,Var0=10/4) ## Density estimation
print(obj)
plot(obj) ## Plot the estimated density
curve(dgamma(x,10,2), ## ... and compare it to the true density (in red)
      add=TRUE,col="red",lwd=2,lty=2)
legend("topright",col=c("black","red"),lwd=c(2,2),lty=c(1,2),
       legend=c("Estimated density","True density"),bty="n")
## Same story for the cdf
with(obj, curve(pdist(x),ymin,ymax,lwd=2,xlab="",ylab="F(x)"))
curve(pgamma(x,10,2),add=TRUE,col="red",lwd=2,lty=2)
legend("right",col=c("black","red"),lwd=c(2,2),lty=c(1,2),
       legend=c("Estimated cdf","True cdf"),bty="n")
```

`densLPS.object` *Object generated by function* `densityLPS`

Description

An object returned by function `densityLPS`: this is a list with various components related to the estimation of a density with given mean and variance from potentially right- or interval-censored data using Laplace P-splines.

Value

An object returned by `densityLPS` has the following elements: Essential part:

- `converged` : logical convergence indicator.
- `ddist` : fitted density function.
- `Hdist` : fitted cumulative hazard function.
- `hdist` : fitted hazard function.
- `pdist` : fitted cumulative distribution function.
- `ymin, ymax` : assumed values for the support of the distribution.
- `phi` : estimated B-spline coefficients for the log-hazard of the error distribution.
- `U.phi` : score of the Lagrangian G($\phi|\omega$).
- `tau, ltau` : selected penalty parameter and its logarithm.
- `est` : vector containing the estimated/selected ($\phi, \log\tau$) parameters.
- `fixed.phi` : logical indicating whether the spline parameters were given fixed values or estimated from the data.
- `phi.ref` : reference values for the spline parameters with respect to which ϕ is compared during penalization.

- `BWB` : Hessian for ϕ without the penalty contribution.
- `Prec` : Hessian or posterior precision matrix for ϕ.
- `Fisher` : Fisher information for ϕ.
- `bins, ugrid, du` : bins (of width 'du') and with midpoints 'ugrid' partitioning the support of the density.
- `h.grid, H.grid, dens.grid` : hazard, cumulative hazard and density values at the grid midpoints 'ugrid'.
- `h.bins, H.bins, dens.bins` : hazard, cumulative hazard and density values at the bin limits 'bins'.
- `expected` : expected number of observations within each bin.
- `Finfty` : integrated density value over the considered support.
- `Mean0, Var0` : when specified, constrained mean and variance values during estimation.
- `mean.dist, var.dist` : mean and variance of the fitted density.
- `method` : method used for penaly selection: "evidence" (by maximizing the marginal posterior for τ) or "Schall" (Schall's method).
- `ed` : effective number of (spline) parameters.
- `iterations` : total number of iterations necessary for convergence.
- `elapsed.time` : time required for convergence.

Additional elements: the content of the Dens1d.object used when densityLPS was called.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

densityLPS, DALSM

logDet.fun *Log-determinant of a positive-definite matrix*

Description

Log-determinant of a positive-definite matrix

Usage

```
logDet.fun(x)
```

Arguments

x	positive definite matrix.

Value

log of det(x).

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

Examples

```
A = matrix(1:4,ncol=2)
logDet.fun(A)
```

Pcal.fun *Generation of the penalty matrix for an additive model based on P-splines*

Description

Compute the penalty matrix associated to a vector containing fixed (non-penalized) parameters and equal-size sub-vectors of penalized B-spline parameters.

Usage

```
Pcal.fun(nfixed, lambda, Pd.x)
```

Arguments

nfixed	the number of fixed (i.e. non-penalized) parameters.
lambda	a vector of p penalty parameters where each component is associated to a sub-vector of spline parameters of length J.
Pd.x	a penalty matrix of size J associated to a given sub-vector of spline parameters.

Value

A block diagonal penalty matrix of size (nfixed+pJ) given by Blockdiag(diag(0,nfixed), diag(lambda).kron.Pd.x).

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

Examples

```
D = diff(diag(5),diff=2) ## Difference penalty matrix of order 2 for 5 P-spline parameters
P = t(D) %*% D ## Penalty matrix of order 2 for 5 B-spline parameters
lambda = c(100,10) ## Penalty parameters for 2 additive terms with 5 B-spline parameters each
Pcal.fun(3,lambda,P) ## Global penalty matrix for 3 additional non-penalized parameters
```

---

plot.DALSM *Plot visual information on a* DALSM.object

Description

Visualize the estimated additive terms and error density corresponding to a Double Additive Location-Scale Model (DALSM) object.

Usage

```
## S3 method for class 'DALSM'
plot(x,
       mfrow.loc=NULL, mfrow.disp=NULL,
       nx=101, equal.ylims=TRUE, true.loc=NULL,true.disp=NULL, ci.level=NULL,
      error.lim = NULL, add.residuals=FALSE, true.derr=NULL, new.dev=TRUE, ...)
```

Arguments

x	a DALSM.object.
mfrow.loc	(optional) window layout to plot the additive terms for location.
mfrow.disp	(optional) window layout to plot the additive terms for dispersion.
nx	(optional) number of points to make the plots for the fitted additive terms (default: 101).
equal.ylims	logical indicating if the same y-limits must be used when plotting the fitted additive terms (default: TRUE).
true.loc	(optional) list of functions to be superposed to the corresponding estimated additive terms in the location submodel (default: NULL).
true.disp	(optional) list of functions to be superposed to the corresponding estimated additive terms in the dispersion submodel (default: NULL).
ci.level	(optional) nominal level for the plotted pointwise credible intervals (default: x$ci.level).
error.lim	(optional) plotting interval for the estimated standardized error density in the DALSM model (default: support of the fitted standardized error density).
add.residuals	logical requesting to add the (possibly censored) standardized residuals to the plot of the fitted standardized error density (default: FALSE).
true.derr	(optional) density function to superpose to the estimated standardized error density when plotting (default: NULL).
new.dev	(optional) logical indicating whether a new plotting device must be opened for each graph (default: TRUE).
...	additional generic plotting arguments.

Details

Plot the fitted additive terms and the estimated standardized error density contained in the DALSM.object x.

Value

In addition to the plots, an invisible list containing the following is returned:

- J1 : number of additive terms in the location sub-model.
- x.loc : a nx by J1 matrix containing a regular grid of nx covariate values where the corresponding additive term in location is evaluated.
- f.loc : a nx by J1 matrix containing the J1 fitted location additive terms evaluated at x.loc.
- se.loc : a nx by J1 matrix containing the the pointwise standard errors of the fitted location additive terms evaluated at x.loc.
- J2 : number of additive terms in the dispersion sub-model.
- x.disp : a nx by J2 matrix containing a regular grid of nx covariate values where the corresponding additive term in dispersion is evaluated.
- f.disp : a nx by J2 matrix containing the J2 fitted dispersion additive terms evaluated at x.disp.
- se.disp : a nx by J2 matrix containing the pointwise standard errors of the fitted dispersion additive terms evaluated at x.disp.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

DALSM, DALSM.object, print.DALSM

Examples

```
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
plot(fit)
```

---

plot.densLPS *Plot the density estimate in a* densLPS.object

Description

Plot the density estimate obtained by densityLPS from censored data with given mean and variance.

Usage

```
## S3 method for class 'densLPS'
plot(x,
       xlim=range(fit$bins),breaks=NULL,hist=FALSE,histRC=FALSE,
       xlab="",ylab="Density",main="",...)
```

Arguments

x	a densLPS.object.
xlim	interval of values where the density should be plotted.
breaks	(Optional) breaks for the histogram of the observed residuals.
hist	Logical (Default: FALSE) indicating whether the histogram of the (pseudo-) data should be plotted with the estimated density.

histRC	Logical (Default: FALSE) indicating whether the histogram of the right-censored residuals should be highlighted.
xlab	Optional label for the x-axis (Defaut: empty).
ylab	Optional label for the y-axis (Default: "Density").
main	Plot main title (Default: "").
...	Optional additional plot parameters.

Value

No returned value (just plots).

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

densLPS.object, print.densLPS, densityLPS.

Examples

```
require(DALSM)

## Example 1: density estimation from simulated IC data
n = 500 ## Sample size
x = 3 + rgamma(n,10,2) ## Exact generated data
width = runif(n,1,3) ## Width of the IC data (mean width = 2)
w = runif(n) ## Positioning of the exact data within the interval
xmat = cbind(x-w*width,x+(1-w)*width) ## Generated IC data
head(xmat)
obj.data = Dens1d(xmat,ymin=0) ## Prepare the data for estimation
## Density estimation with fixed mean and variance
obj = densityLPS(obj.data,Mean0=3+10/2,Var0=10/4)
plot(obj, hist=TRUE) ## Histogram of the pseudo-data with the density estimate
curve(dgamma(x-3,10,2), ## ... compared to the true density (in red)
      add=TRUE,col="red",lwd=2,lty=2)
legend("topright",col=c("black","red","grey"),lwd=c(2,2,10),lty=c(1,2,1),
       legend=c("Fitted density","True density","Pseudo-data"),bty="n")
print(obj) ## ... with summary statistics

## Example 2: estimation of the error density in a DALSM model
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
```

```
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
plot(fit$derr, hist=TRUE)  ## Plot the estimated error density
legend("topright",col=c("black","grey"),lwd=c(2,10),lty=c(1,1),
       legend=c("Estimated error density","Pseudo-residuals"),bty="n")
print(fit$derr) ## ... and provide summary statistics for it
```

---

plotRegion *Plot a credible region for a curve together with its point estimates*

Description

Plot a credible region (in grey) for a curve together with its point estimates.

Usage

```
plotRegion(x,mat,
       add=FALSE, xlim=range(x), ylim=range(mat),
       lwd=2, xlab="", ylab="", main="", ...)
```

Arguments

x	vector of values where the curve is evaluated.
mat	cbind(f.hat,f.low,f.up) is a matrix containing the point estimates <f.hat> and the liming values <f.low> and <f.up> for the credible region.
add	Logical indicating if the plot must be added to the active plot (Default: FALSE).
xlim	range of <x> values for which the plot should be provided.
ylim	range of curve values that should be considered for the plot.
lwd	line width for the plot (Default: 2).
xlab	x-label.
ylab	y-label.
main	main title.
...	optional plotting parameters.

Value

No returned value (just a plot)

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

DALSM, DALSM.object.

Examples

```
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
obj = DALSM_additive(fit)
## par(mfrow=c(1,2),mar=c(4,5,1,1))
with(obj$f.loc.grid$age, plotRegion(x, y.mat,
     xlab="age", ylab=expression('f'[1]^{~mu}*(age))))
with(obj$f.disp.grid$age, plotRegion(x, y.mat,
     xlab="age", ylab=expression('f'[1]^{~sigma}*(age))))
```

---

predict.DALSM *Prediction based on a DALSM model*

Description

Estimated conditional mean and standard deviation of the response based on a DALSM object for given covariate values in a data frame 'newdata'. Conditional quantiles can also be computed.

Usage

```
## S3 method for class 'DALSM'
predict(object, newdata, probs, ...)
```

Arguments

object	a DALSM.object.
newdata	an optional data frame in which to look for variables with which to predict. If omitted, the covariate values in the original data frame used to fit the DALSM model are considered.
probs	probability levels of the requested conditional quantiles.
...	further arguments passed to or from other methods.

Value

Returns a list containing:

- mu : estimated conditional mean.
- sd : estimated conditional standard deviation.
- quant : estimated quantiles (at probability level probs) of the fitted conditional response in the DALSM model.
- qerr : quantiles (at probability level probs) of the fitted error distribution in the DALSM model.
- probs : a reminder of the requested probability levels for the fitted quantiles.

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

DALSM.object, print.DALSM, plot.DALSM.

Examples

```
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
data2 = data.frame(age=c(40,60),eduyrs=c(18,12))
predict(fit, data = DALSM_IncomeData, newdata=data2, probs=c(.2,.5,.8))
```

---

print.DALSM *Print summary information on a* DALSM.object

Description

Print summary information on a DALSM.object.

Usage

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

Arguments

x	an object of class DALSM.object.
...	additional generic printing arguments.

Details

Provides summary measures on the estimation of the regression parameters and additive terms in the location and dispersion submodels corresponding to a DALSM.object generated by DALSM.

Value

No returned value (just printed summary).

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

plot.DALSM, DALSM.object, DALSM.

Examples

```
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
print(fit)
```

---

print.densLPS *Print a summary of the information in a* densLPS.object

---

Description

Print summary information on the density estimate obtained by densityLPS from censored data with given mean and variance.

Usage

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

Arguments

x	a densLPS.object.
...	Optional additional print parameters.

Value

No returned value (just printed summary).

Author(s)

Philippe Lambert <p.lambert@uliege.be>

References

Lambert, P. (2021). Fast Bayesian inference using Laplace approximations in nonparametric double additive location-scale models with right- and interval-censored data. *Computational Statistics and Data Analysis*, 161: 107250. <doi:10.1016/j.csda.2021.107250>

See Also

densLPS.object, plot.densLPS, densityLPS.

Examples

```
require(DALSM)
data(DALSM_IncomeData)
resp = DALSM_IncomeData[,1:2]
fit = DALSM(y=resp,
            formula1 = ~twoincomes+s(age)+s(eduyrs),
            formula2 = ~twoincomes+s(age)+s(eduyrs),
            data = DALSM_IncomeData)
plot(fit$derr)  ## Plot the estimated error density
print(fit$derr) ## ... and provide some descriptive elements on it
```

---

qknots *Specification of knots in a cubic P-spline model*

Description

Specification of knots for a cubic B-spline basis with K elements in a P-spline model. The knots should support the data contained in vector x and are by default assumed equidistant. Alternatively, they can be based on the data quantiles. The penalty matrix of the selected penalty order (3 by default) is also returned.

Usage

```
qknots(x, xmin = NULL, xmax = NULL, equid.knots = TRUE, pen.order = 3, K = 25)
```

Arguments

x	data that the knots should upport.
xmin	desired minimum value for the knots.
xmax	desired maximum value for the knots.
equid.knots	logical indicating if equidistant knots are desired (Default: TRUE). If FALSE, the quantile of x are used to select the knots.
pen.order	penalty order if equid.knots is TRUE.
K	number of B-splines in the basis.

Value

A list containing:

- xmin : specified minimum value for the knots, except if $\min(x) < xmin$, in which case the default value $\min(x) - sd(x)$ is returned.
- xmin : specified maximum value for the knots, except if $xmax < \max(x)$, in which case the default value $\max(x) + sd(x)$ is returned.
- K : number of B-splines.
- knots : equidistant knots on (xmin,xmax) if equidistant.knots is TRUE, based on quantiles of x otherwise.
- Pd : $K \times K$ penalty matrix of order pen.order.
- pen.order : a reminder of the requested penalty order (Default: 3).

Examples

```
x = runif(500)
obj = qknots(x,xmin=0,xmax=1,K=13)
print(obj)
```

Index

∗ **datasets**
DALSM_IncomeData, 10

centeredBasis.gen, 2

DALSM, 3, *5, 6*, *8, 9*, *17*, *21*, *24*, *26*
DALSM.object, *5*, 5, *8, 9*, *20, 21*, *24–26*
DALSM_additive, 8
DALSM_IncomeData, 10
Dens1d, 11, *12–15*
Dens1d.object, *11, 12*, 12, *15*, *17*
densityLPS, *8*, *11–13*, 14, *16, 17*, *22*, *27*
densLPS.object, *5*, *8*, *13*, *15*, 16, *21, 22*, *27*
DesignFormula, *6*

logDet.fun, 18

Pcal.fun, 18
plot.DALSM, *5*, *8, 9*, 19, *25, 26*
plot.densLPS, *15*, 21, *27*
plotRegion, 23
predict.DALSM, 24
print.DALSM, *5*, *8, 9*, *21*, *25*, 25
print.densLPS, *15*, *22*, 26

qknots, 27