

Package ‘gss’

April 17, 2009

Version 1.0-5

Depends R (>= 1.7.0)

Title General Smoothing Splines

Author Chong Gu <chong@stat.purdue.edu>

Maintainer Chong Gu <chong@stat.purdue.edu>

Description A comprehensive package for structural multivariate function estimation using smoothing splines.

License GPL

Repository CRAN

Date/Publication 2009-03-10 07:41:54

R topics documented:

aids	2
bacteriuria	2
buffalo	3
cdssden	4
dssden	5
fitted.ssanova	6
gastric	6
gauss.quad	7
gssanova	7
gssanova0	10
hzdrate.sshzd	13
LakeAcidity	14
nlm0	15
nox	15
ozone	17
predict.ssanova	18
print	20

project	21
smolyak	22
ssanova	23
ssanova0	25
ssden	27
sshzd	30
stan	33
summary.gssanova	33
summary.gssanova0	35
summary.ssanova	36
wesdr	37

Index 39

aids	<i>AIDS Incubation</i>
------	------------------------

Description

A data set collected by Centers for Disease Control and Prevention concerning AIDS patients who were infected with the HIV virus through blood transfusion.

Usage

`data(aids)`

Format

A data frame containing 295 observations on the following variables.

<code>incu</code>	Time from HIV infection to AIDS diagnosis.
<code>infe</code>	Time from HIV infection to end of data collection (July 1986).
<code>age</code>	Age at time of blood transfusion.

Source

Wang, M.-C. (1989), A semiparametric model for randomly truncated data. *Journal of the American Statistical Association*, **84**, 742–748.

bacteriuria	<i>Treatment of Bacteriuria</i>
-------------	---------------------------------

Description

Bacteriuria patients were randomly assigned to two treatment groups. Weekly binary indicator of bacteriuria was recorded for every patient over 4 to 16 weeks. A total of 72 patients were represented in the data, with 36 each in the two treatment groups.

Usage

```
data(bacteriuria)
```

Format

A data frame containing 820 observations on the following variables.

id	Identification of patients, a factor.
trt	Treatments 1 or 2, a factor.
time	Weeks after randomization.
infect	Binary indicator of bacteriuria (bacteria in urine).

Source

Joe, H. (1997), *Multivariate Models and Dependence Concepts*. London: Chapman and Hall.

References

Gu, C. and Ma, P. (2005), Generalized nonparametric mixed-effect models: computation and smoothing parameter selection. *Journal of Computational and Graphical Statistics*, **14**, 485–504.

buffalo

Buffalo Annual Snowfall

Description

Annual snowfall accumulations in Buffalo, NY from 1910 to 1973.

Usage

```
data(buffalo)
```

Format

A vector of 64 numerical values.

Source

Scott, D. W. (1985), Average shifted histograms: Effective nonparametric density estimators in several dimensions. *The Annals of Statistics*, **13**, 1024–1040.

cdssden

*Evaluating Conditional PDF, CDF, and Quantiles of Smoothing Spline
Density Estimates*

Description

Evaluate conditional pdf, cdf, and quantiles for smoothing spline density estimates.

Usage

```
cdssden(object, x, cond, int=NULL)
cpssden(object, q, cond, int=NULL)
cqssden(object, p, cond, int=NULL)
```

Arguments

object	Object of class "ssden".
x	Data frame or vector of points on which conditional density is to be evaluated.
cond	One row data frame of conditioning variables.
int	Normalizing constant.
q	Vector of points on which conditional cdf is to be evaluated.
p	Vector of probabilities for which conditional quantiles are to be calculated.

Details

The argument `x` in `cdssden` is of the same form as the argument `newdata` in `predict.lm`, but can take a vector for 1-D conditional densities.

`cpssden` and `cqssden` naturally only work for 1-D conditional densities of a numerical variable.

Value

`cdssden` returns a list object with the following components.

pdf	Vector of conditional pdf.
int	Normalizing constant.

`cpssden` and `cqssden` return a vector of conditional cdf or quantiles.

Note

If variables other than factors or numerical vectors are involved in `x`, the normalizing constant can not be computed.

`cpssden` and `cqssden` can be very slow.

See Also

Fitting function `ssden` and `dssden`.

dssden	<i>Evaluating PDF, CDF, and Quantiles of Smoothing Spline Density Estimates</i>
--------	---

Description

Evaluate pdf, cdf, and quantiles for smoothing spline density estimates.

Usage

```
dssden(object, x)
pssden(object, q)
qssden(object, p)
```

Arguments

object	Object of class "ssden".
x	Data frame or vector of points on which density is to be evaluated.
q	Vector of points on which cdf is to be evaluated.
p	Vector of probabilities for which quantiles are to be calculated.

Details

The argument `x` in `dssden` is of the same form as the argument `newdata` in `predict.lm`, but can take a vector for 1-D densities.

`pssden` and `qssden` naturally only work for 1-D densities.

Value

A vector of pdf, cdf, or quantiles.

See Also

Fitting function `ssden` and `cdssden`.

<code>fitted.ssanova</code>	<i>Fitted Values and Residuals from Smoothing Spline ANOVA Fits</i>
-----------------------------	---

Description

Methods for extracting fitted values and residuals from smoothing spline ANOVA fits.

Usage

```
## S3 method for class 'ssanova':
fitted(object, ...)
## S3 method for class 'ssanova':
residuals(object, ...)

## S3 method for class 'gssanova':
fitted(object, ...)
## S3 method for class 'gssanova':
residuals(object, type="working", ...)
```

Arguments

<code>object</code>	Object of class "ssanova" or "gssanova".
<code>type</code>	Type of residuals desired, with two alternatives "working" (default) or "deviance".
<code>...</code>	Ignored.

Details

The fitted values for "gssanova" objects are on the link scale, so are the "working" residuals.

<code>gastric</code>	<i>Gastric Cancer Data</i>
----------------------	----------------------------

Description

Survival of gastric cancer patients under chemotherapy and chemotherapy-radiotherapy combination.

Usage

```
data(gastric)
```

Format

A data frame containing 90 observations on the following variables.

<code>futime</code>	Follow-up time, in days.
<code>status</code>	Censoring status.
<code>trt</code>	Factor indicating the treatments: 1 – chemotherapy, 2 – combination.

Source

Moreau, T., O'Quigley, J., and Mesbah, M. (1985), A global goodness-of-fit statistic for the proportional hazards model. *Applied Statistics*, **34**, 212-218.

 gauss.quad

Generating Gauss-Legendre Quadrature

Description

Generate Gauss-Legendre quadratures using the FORTRAN routine `gaussq.f` found on NETLIB.

Usage

```
gauss.quad(size, interval)
```

Arguments

<code>size</code>	Size of quadrature.
<code>interval</code>	Interval to be covered.

Value

`gauss.quad` returns a list object with the following components.

<code>pt</code>	Quadrature nodes.
<code>wt</code>	Quadrature weights.

 gssanova

Fitting Smoothing Spline ANOVA Models with Non-Gaussian Responses

Description

Fit smoothing spline ANOVA models in non-Gaussian regression. The symbolic model specification via `formula` follows the same rules as in `lm` and `glm`.

Usage

```
gssanova(formula, family, type=NULL, data=list(), weights, subset,
          offset, na.action=na.omit, partial=NULL, alpha=NULL, nu=NULL,
          id.basis=NULL, nbasis=NULL, seed=NULL, random=NULL)
```

Arguments

<code>formula</code>	Symbolic description of the model to be fit.
<code>family</code>	Description of the error distribution. Supported are exponential families "binomial", "poisson", "Gamma", and "nbinomial". Also supported are accelerated life model families "weibull", "lognorm", and "loglogis".
<code>type</code>	List specifying the type of spline for each variable. See mkterm for details.
<code>data</code>	Optional data frame containing the variables in the model.
<code>weights</code>	Optional vector of weights to be used in the fitting process.
<code>subset</code>	Optional vector specifying a subset of observations to be used in the fitting process.
<code>offset</code>	Optional offset term with known parameter 1.
<code>na.action</code>	Function which indicates what should happen when the data contain NAs.
<code>partial</code>	Optional extra unpenalized terms in partial spline models.
<code>alpha</code>	Tuning parameter defining cross-validation; larger values yield smoother fits. Defaults are <code>alpha=1</code> for <code>family="binomial"</code> and <code>alpha=1.4</code> otherwise.
<code>nu</code>	Inverse scale parameter in accelerated life model families. Ignored for exponential families.
<code>id.basis</code>	Index designating selected "knots".
<code>nbasis</code>	Number of "knots" to be selected. Ignored when <code>id.basis</code> is supplied.
<code>seed</code>	Seed for reproducible random selection of "knots". Ignored when <code>id.basis</code> is supplied.
<code>random</code>	Input for parametric random effects in nonparametric mixed-effect models. See mkran for details.

Details

The model specification via `formula` is intuitive. For example, `y~x1*x2` yields a model of the form

$$y = C + f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2) + e$$

with the terms denoted by "1", "x1", "x2", and "x1:x2".

The model terms are sums of unpenalized and penalized terms. Attached to every penalized term there is a smoothing parameter, and the model complexity is largely determined by the number of smoothing parameters.

Only one link is implemented for each family. It is the logit link for "binomial", and the log link for "poisson", and "Gamma". For "nbinomial", the working parameter is the logit of the probability p ; see [NegBinomial](#). For "weibull", "lognorm", and "loglogis", it is the location parameter for the log lifetime.

The selection of smoothing parameters is through direct cross-validation. The cross-validation score used for `family="poisson"` is taken from density estimation as in Gu and Wang (2003), and those used for other families are derived following the lines of Gu and Xiang (2001).

A subset of the observations are selected as "knots." Unless specified via `id.basis` or `nbasis`, the number of "knots" q is determined by $\max(30, 10n^{2/9})$, which is appropriate for the default cubic splines for numerical vectors.

Value

`gssanova` returns a list object of class `c("gssanova", "ssanova")`.

The method `summary.gssanova` can be used to obtain summaries of the fits. The method `predict.ssanova` can be used to evaluate the fits at arbitrary points along with standard errors, on the link scale. The method `project.gssanova` can be used to calculate the Kullback-Leibler projection for model selection. The methods `residuals.gssanova` and `fitted.gssanova` extract the respective traits from the fits.

Responses

For `family="binomial"`, the response can be specified either as two columns of counts or as a column of sample proportions plus a column of total counts entered through the argument `weights`, as in `glm`.

For `family="nbinomial"`, the response may be specified as two columns with the second being the known sizes, or simply as a single column with the common unknown size to be estimated through the maximum likelihood.

For `family="weibull"`, `"lognorm"`, or `"loglogis"`, the response consists of three columns, with the first giving the follow-up time, the second the censoring status, and the third the left-truncation time. For data with no truncation, the third column can be omitted.

Note

For simpler models and moderate sample sizes, the exact solution of `gssanova0` can be faster.

The results may vary from run to run. For consistency, specify `id.basis` or set `seed`.

In `gss` versions earlier than 1.0, `gssanova` was under the name `gssanova1`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. and Xiang, D. (2001), Cross validating non Gaussian data: generalized approximate cross validation revisited. *Journal of Computational and Graphical Statistics*, **10**, 581–591.

Gu, C. and Wang, J. (2003), Penalized likelihood density estimation: Direct cross-validation and scalable approximation. *Statistica Sinica*, **13**, 811–826.

Examples

```
## Fit a cubic smoothing spline logistic regression model
test <- function(x)
  {.3*(1e6*(x^11*(1-x)^6)+1e4*(x^3*(1-x)^10))-2}
x <- (0:100)/100
p <- 1-1/(1+exp(test(x)))
y <- rbinom(x, 3, p)
logit.fit <- gssanova(cbind(y, 3-y)~x, family="binomial")
## The same fit
logit.fit1 <- gssanova(y/3~x, "binomial", weights=rep(3, 101),
```

```

                                id.basis=logit.fit$id.basis)
## Obtain estimates and standard errors on a grid
est <- predict(logit.fit,data.frame(x=x),se=TRUE)
## Plot the fit and the Bayesian confidence intervals
plot(x,y/3,ylab="p")
lines(x,p,col=1)
lines(x,1-1/(1+exp(est$fit)),col=2)
lines(x,1-1/(1+exp(est$fit+1.96*est$se)),col=3)
lines(x,1-1/(1+exp(est$fit-1.96*est$se)),col=3)

## Fit a mixed-effect logistic model
data(bacteriuria)
bact.fit <- gssanova(infect~trt+time,family="binomial",data=bacteriuria,
                    id.basis=(1:820)[bacteriuria$id%in%c(3,38)],random=~1|id)
## Predict fixed effects
predict(bact.fit,data.frame(time=2:16,trt=as.factor(rep(1,15))),se=TRUE)
## Estimated random effects
bact.fit$b

## Clean up
## Not run:
rm(test,x,p,y,logit.fit,logit.fit1,est,bacteriuria,bact.fit)
dev.off()
## End(Not run)

```

gssanova0

Fitting Smoothing Spline ANOVA Models with Non-Gaussian Responses

Description

Fit smoothing spline ANOVA models in non-Gaussian regression. The symbolic model specification via formula follows the same rules as in [lm](#) and [glm](#).

Usage

```

gssanova0(formula, family, type=NULL, data=list(), weights, subset,
           offset, na.action=na.omit, partial=NULL, method=NULL,
           varht=1, nu=NULL, prec=1e-7, maxiter=30)

```

Arguments

formula	Symbolic description of the model to be fit.
family	Description of the error distribution. Supported are exponential families "binomial", "poisson", "Gamma", "inverse.gaussian", and "nbinomial". Also supported are accelerated life model families "weibull", "lognorm", and "loglogis".
type	List specifying the type of spline for each variable. See mkterm for details.

<code>data</code>	Optional data frame containing the variables in the model.
<code>weights</code>	Optional vector of weights to be used in the fitting process.
<code>subset</code>	Optional vector specifying a subset of observations to be used in the fitting process.
<code>offset</code>	Optional offset term with known parameter 1.
<code>na.action</code>	Function which indicates what should happen when the data contain NAs.
<code>partial</code>	Optional extra unpenalized terms in partial spline models.
<code>method</code>	Score used to drive the performance-oriented iteration. Supported are <code>method="v"</code> for GCV, <code>method="m"</code> for GML, and <code>method="u"</code> for Mallows's CL.
<code>varht</code>	Dispersion parameter needed for <code>method="u"</code> . Ignored when <code>method="v"</code> or <code>method="m"</code> are specified.
<code>nu</code>	Inverse scale parameter in accelerated life model families. Ignored for exponential families.
<code>prec</code>	Precision requirement for the iterations.
<code>maxiter</code>	Maximum number of iterations allowed for performance-oriented iteration, and for inner-loop multiple smoothing parameter selection when applicable.

Details

The model specification via `formula` is intuitive. For example, `y~x1*x2` yields a model of the form

$$y = C + f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2) + e$$

with the terms denoted by `"1"`, `"x1"`, `"x2"`, and `"x1:x2"`.

The model terms are sums of unpenalized and penalized terms. Attached to every penalized term there is a smoothing parameter, and the model complexity is largely determined by the number of smoothing parameters.

Only one link is implemented for each family. It is the logit link for `"binomial"`, and the log link for `"poisson"`, `"Gamma"`, and `"inverse.gaussian"`. For `"nbinomial"`, the working parameter is the logit of the probability p ; see [NegBinomial](#). For `"weibull"`, `"lognorm"`, and `"loglogis"`, it is the location parameter for the log lifetime.

The models are fitted by penalized likelihood method through the performance-oriented iteration as described in the reference; the $O(n^3)$ algorithms of RKPACk are used for numerical calculations. For `family="binomial"`, `"poisson"`, `"nbinomial"`, `"weibull"`, `"lognorm"`, and `"loglogis"`, the score driving the performance-oriented iteration defaults to `method="u"` with `varht=1`. For `family="Gamma"` and `"inverse.gaussian"`, the default is `method="v"`.

Value

`gssanova0` returns a list object of class `c("gssanova0", "ssanova0", "gssanova")`.

The method `summary.gssanova0` can be used to obtain summaries of the fits. The method `predict.ssanova0` can be used to evaluate the fits at arbitrary points along with standard errors, on the link scale. The methods `residuals.gssanova` and `fitted.gssanova` extract the respective traits from the fits.

Responses

For `family="binomial"`, the response can be specified either as two columns of counts or as a column of sample proportions plus a column of total counts entered through the argument `weights`, as in `glm`.

For `family="nbinomial"`, the response may be specified as two columns with the second being the known sizes, or simply as a single column with the common unknown size to be estimated through the maximum likelihood.

For `family="weibull"`, `"lognorm"`, or `"loglogis"`, the response consists of three columns, with the first giving the follow-up time, the second the censoring status, and the third the left-truncation time. For data with no truncation, the third column can be omitted.

Note

The direct cross-validation of `gssanova` can be more effective in general, and more stable for complex models.

For large sample sizes, the approximate solution of `gssanova` can be faster.

The method `project` is not implemented for `gssanova0`, nor is the mixed-effect model support through `mkran`.

In `gss` versions earlier than 1.0, `gssanova0` was under the name `gssanova`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (1992), Cross-validating non Gaussian data. *Journal of Computational and Graphical Statistics*, **1**, 169-179.

Examples

```
## Fit a cubic smoothing spline logistic regression model
test <- function(x)
  {.3*(1e6*(x^11*(1-x)^6)+1e4*(x^3*(1-x)^10))-2}
x <- (0:100)/100
p <- 1-1/(1+exp(test(x)))
y <- rbinom(x, 3, p)
logit.fit <- gssanova0(cbind(y, 3-y)~x, family="binomial")
## The same fit
logit.fit1 <- gssanova0(y/3~x, "binomial", weights=rep(3, 101))
## Obtain estimates and standard errors on a grid
est <- predict(logit.fit, data.frame(x=x), se=TRUE)
## Plot the fit and the Bayesian confidence intervals
plot(x, y/3, ylab="p")
lines(x, p, col=1)
lines(x, 1-1/(1+exp(est$fit)), col=2)
lines(x, 1-1/(1+exp(est$fit+1.96*est$se)), col=3)
lines(x, 1-1/(1+exp(est$fit-1.96*est$se)), col=3)
## Clean up
```

```
## Not run:
rm(test,x,p,y,logit.fit,logit.fit1,est)
dev.off()
## End(Not run)
```

hzdrate.sshzd

Evaluating Smoothing Spline Hazard Estimates

Description

Evaluate smoothing spline hazard estimates by `sshzd`.

Usage

```
hzdrate.sshzd(object, x, se=FALSE)
hzdcurve.sshzd(object, time, covariates=NULL, se=FALSE)
survexp.sshzd(object, time, covariates=NULL, start=0)
```

Arguments

<code>object</code>	Object of class "sshzd".
<code>x</code>	Data frame or vector of points on which hazard is to be evaluated.
<code>se</code>	Flag indicating if standard errors are required.
<code>time</code>	Vector of time points.
<code>covariates</code>	Vector of covariate values.
<code>start</code>	Optional starting times of the intervals.

Value

For `se=FALSE`, `hzdrate.sshzd` returns a vector of hazard evaluations, and `hzdcurve.sshzd` returns a vector or columns of hazard curve(s) evaluated on `time` points at the `covariates` values. For `se=TRUE`, `hzdrate.sshzd` and `hzdcurve.sshzd` return a list consisting of the following components.

<code>fit</code>	Vector or columns of hazard.
<code>se.fit</code>	Vector or columns of standard errors for log hazard.

`survexp.sshzd` returns a vector of expected survivals based on the cumulative hazards over $(start, time)$, which in fact are the (conditional) survival probabilities $S(time)/S(start)$.

Note

For left-truncated data, `start` must be at or after the earliest truncation point.

See Also

Fitting function [sshzd](#).

LakeAcidity *Water Acidity in Lakes*

Description

Data extracted from the Eastern Lake Survey of 1984 conducted by the United States Environmental Protection Agency, concerning 112 lakes in the Blue Ridge.

Usage

```
data(LakeAcidity)
```

Format

A list containing 112 observations on the following variables.

ph	Surface ph.
cal	Calcium concentration.
lat	Latitude.
lon	Longitude.
geog	Geographic location, derived from lat and lon

Details

geog was generated from lat and lon using the code given in the Example section.

Source

Douglas, A. and Delampady, M. (1990), *Eastern Lake Survey – Phase I: Documentation for the Data Base and the Derived Data sets*. Tech Report 160 (SIMS), Dept. Statistics, University of British Columbia.

References

Gu, C. and Wahba, G. (1993), Semiparametric analysis of variance with tensor product thin plate splines. *Journal of the Royal Statistical Society Ser. B*, **55**, 353–368.

Examples

```
## Converting latitude and longitude to x-y coordinates
## Not run:
convert <- function(lat, lon) {
  lat <- lat/180*pi
  lon <- lon/180*pi
  m.lat <- (max(lat)+min(lat))/2
  m.lon <- (max(lon)+min(lon))/2
  x <- cos(m.lat)*sin(m.lon-lon)
  y <- sin(lat-m.lat)
```

```

      cbind(x,y)
    }
  data(LakeAcidity)
  convert(LakeAcidity$lat,LakeAcidity$lon)
  ## Clean up
  rm(LakeAcidity,convert)
  ## End(Not run)

```

nlm0

Minimizing Univariate Functions on Finite Intervals

Description

Minimize univariate functions on finite intervals using 3-point quadratic fit, with golden-section safe-guard.

Usage

```
nlm0(fun, range, prec=1e-7)
```

Arguments

fun	Function to be minimized.
range	Interval on which the function to be minimized.
prec	Desired precision of the solution.

Value

nlm0 returns a list object with the following components.

estimate	Minimizer.
minimum	Minimum.
evaluations	Number of function evaluations.

nox

NOx in Engine Exhaust

Description

Data from an experiment in which a single-cylinder engine was run with ethanol to see how the NOx concentration in the exhaust depended on the compression ratio and the equivalence ratio.

Usage

```
data(nox)
```

Format

A data frame containing 88 observations on the following variables.

nox NOx concentration in exhaust.
 comp Compression ratio.
 equi Equivalence ratio.

Source

Brinkman, N. D. (1981), Ethanol fuel – a single-cylinder engine study of efficiency and exhaust emissions. *SAE Transactions*, **90**, 1410–1424.

References

Cleveland, W. S. and Devlin, S. J. (1988), Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, **83**, 596–610.
 Breiman, L. (1991), The pi method for estimating multivariate functions from noisy data. *Technometrics*, **33**, 125–160.

ozone

Ozone Concentration in Los Angeles Basin

Description

Daily measurements of ozone concentration and eight meteorological quantities in the Los Angeles basin for 330 days of 1976.

Usage

data(ozone)

Format

A data frame containing 330 observations on the following variables.

upo3	Upland ozone concentration, in ppm.
vdht	Vandenberg 500 millibar height, in meters.
wdsp	Wind speed, in miles per hour.
hmdt	Humidity.
sbtp	Sandburg Air Base temperature, in Celsius.
ibht	Inversion base height, in foot.
dgpg	Dagget pressure gradient, in mmHg.
ibtp	Inversion base temperature, in Fahrenheit.
vsty	Visibility, in miles.
day	Calendar day, between 1 and 366.

Source

Unknown.

References

- Breiman, L. and Friedman, J. H. (1985), Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, **80**, 580–598.
- Hastie, T. and Tibshirani, R. (1990), *Generalized Additive Models*. Chapman and Hall.

predict.ssanova *Predicting from Smoothing Spline ANOVA Fits*

Description

Evaluate terms in a smoothing spline ANOVA fit at arbitrary points. Standard errors of the terms can be requested for use in constructing Bayesian confidence intervals.

Usage

```
## S3 method for class 'ssanova':
predict(object, newdata, se.fit=FALSE,
        include=object$terms$labels, ...)
## S3 method for class 'ssanova0':
predict(object, newdata, se.fit=FALSE,
        include=object$terms$labels, ...)
```

Arguments

object	Object of class inheriting from "ssanova".
newdata	Data frame or model frame in which to predict.
se.fit	Flag indicating if standard errors are required.
include	List of model terms to be included in the prediction. The <code>partial</code> and <code>offset</code> terms, if present, are to be specified by "partial" and "offset", respectively.
...	Ignored.

Value

For `se.fit=FALSE`, `predict.ssanova` returns a vector of the evaluated fit.

For `se.fit=TRUE`, `predict.ssanova` returns a list consisting of the following components.

fit	Vector of evaluated fit.
se.fit	Vector of standard errors.

Note

To supply the partial terms for partial spline models, add a component `partial=I(...)` in `newdata`; the "as is" function `I(...)` is necessary when `partial` has more than one column.

For mixed-effect models through `ssanova` or `gssanova`, the Z matrix is set to 0 if not supplied. To supply the Z matrix, add a component `random=I(...)` in `newdata`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (1992), Penalized likelihood regression: a Bayesian analysis. *Statistica Sinica*, **2**, 255–264.

Gu, C. and Wahba, G. (1993), Smoothing spline ANOVA with component-wise Bayesian "confidence intervals." *Journal of Computational and Graphical Statistics*, **2**, 97–117.

Kim, Y.-J. and Gu, C. (2004), Smoothing spline Gaussian regression: more scalable computation via efficient approximation. *Journal of the Royal Statistical Society, Ser. B*, **66**, 337–356.

See Also

Fitting functions [ssanova](#), [ssanova0](#), [gssanova](#), [gssanova0](#) and methods [summary.ssanova](#), [summary.gssanova](#), [summary.gssanova0](#), [project.ssanova](#), [fitted.ssanova](#).

Examples

```
## THE FOLLOWING EXAMPLE IS TIME-CONSUMING
## Not run:
## Fit a model with cubic and thin-plate marginals, where geog is 2-D
data(LakeAcidity)
fit <- ssanova(ph~log(cal)*geog,,LakeAcidity)
## Obtain estimates and standard errors on a grid
new <- data.frame(cal=1,geog=I(matrix(0,1,2)))
new <- model.frame(~log(cal)+geog,new)
predict(fit,new,se=TRUE)
## Evaluate the geog main effect
predict(fit,new,se=TRUE,inc="geog")
## Evaluate the sum of the geog main effect and the interaction
predict(fit,new,se=TRUE,inc=c("geog","log(cal):geog"))
## Evaluate the geog main effect on a grid
grid <- seq(-.04,.04,len=21)
new <- model.frame(~geog,list(geog=cbind(rep(grid,21),rep(grid,rep(21,21))))))
est <- predict(fit,new,se=TRUE,inc="geog")
## Plot the fit and standard error
par(pty="s")
contour(grid,grid,matrix(est$fit,21,21),col=1)
contour(grid,grid,matrix(est$se,21,21),add=TRUE,col=2)
## Clean up
rm(LakeAcidity,fit,new,grid,est)
dev.off()
## End(Not run)
```

print

Print Functions for Smoothing Spline ANOVA Models

Description

Print functions for Smoothing Spline ANOVA models.

Usage

```
## S3 method for class 'ssanova':
print(x, ...)
## S3 method for class 'ssanova0':
print(x, ...)
## S3 method for class 'gssanova':
print(x, ...)
## S3 method for class 'ssden':
print(x, ...)
## S3 method for class 'sshzd':
print(x, ...)
## S3 method for class 'summary.ssanova':
print(x, digits=6, ...)
## S3 method for class 'summary.gssanova':
print(x, digits=6, ...)
## S3 method for class 'summary.gssanova0':
print(x, digits=6, ...)
```

Arguments

x	Object of class <code>ssanova</code> , <code>summary.ssanova</code> , <code>summary.gssanova</code> , or <code>ssden</code> .
digits	Number of significant digits to be printed in values.
...	Ignored.

See Also

[ssanova](#), [ssanova0](#), [gssanova](#), [gssanova0](#), [ssden](#), [sshzd](#), [summary.ssanova](#), [summary.gssanova](#), [summary.gssanova0](#).

Description

Calculate Kullback-Leibler projection of smoothing spline ANOVA fits for model diagnostics.

Usage

```
project(object, ...)
## S3 method for class 'ssanova':
project(object, include, ...)
## S3 method for class 'gssanova':
project(object, include, ...)
## S3 method for class 'ssden':
project(object, include, mesh=FALSE, ...)
## S3 method for class 'sshzd':
project(object, include, mesh=FALSE, ...)
```

Arguments

object	Object of class "ssanova", "gssanova", "ssden", or "sshzd".
...	Additional arguments. Ignored in <code>project.x</code> .
include	List of model terms to be included in the reduced model space. The <code>partial</code> and <code>offset</code> terms, if present, are to be specified by "partial" and "offset", respectively.
mesh	Flag indicating whether to return evaluations of the projection.

Details

The entropy $KL(\text{fit0}, \text{null})$ can be decomposed as the sum of $KL(\text{fit0}, \text{fit1})$ and $KL(\text{fit1}, \text{null})$, where `fit0` is the fit to be projected, `fit1` is the projection in the reduced model space, and `null` is the constant fit. The ratio $KL(\text{fit0}, \text{fit1})/KL(\text{fit0}, \text{null})$ serves as a diagnostic of the feasibility of the reduced model.

For regression fits, smoothness safe-guard is used to prevent interpolation, and $KL(\text{fit0}, \text{fit1}) + KL(\text{fit1}, \text{null})$ may not match $KL(\text{fit0}, \text{null})$ perfectly.

For mixed-effect models from `ssanova` and `gssanova`, the estimated random effects are treated as `offset`.

Value

The functions return a list consisting of the following components.

ratio	$KL(\text{fit0}, \text{fit1})/KL(\text{fit0}, \text{null})$; the smaller the value, the more feasible the reduced model is.
kl	$KL(\text{fit0}, \text{fit1})$.

check	$KL(\text{fit0}, \text{fit1})/KL(\text{fit0}, \text{null}) + KL(\text{fit1}, \text{null})/KL(\text{fit0}, \text{null})$; a value closer to 1 is preferred.
mesh	The evaluations of the projection.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (2004), Model diagnostics for smoothing spline ANOVA models. *The Canadian Journal of Statistics*, **32**, 347–358.

See Also

Fitting functions [ssanova](#), [gssanova](#), [ssden](#), and [sshzd](#).

 smolyak

Generating Smolyak Cubature

Description

Generate delayed Smolyak cubatures using C routines modified from `smolyak.c` found in Knut Petras' SMOLPACK.

Usage

```
smolyak.quad(d, k)
```

```
smolyak.size(d, k)
```

Arguments

d Dimension of unit cube.

k Depth of algorithm.

Value

`smolyak.quad` returns a list object with the following components.

pt Quadrature nodes in rows of matrix.

wt Quadrature weights.

`smolyak.size` returns an integer.

Description

Fit smoothing spline ANOVA models in Gaussian regression. The symbolic model specification via formula follows the same rules as in [lm](#).

Usage

```
ssanova(formula, type=NULL, data=list(), weights, subset, offset,
         na.action=na.omit, partial=NULL, method="v", alpha=1.4,
         varht=1, id.basis=NULL, nbasis=NULL, seed=NULL, random=NULL)
```

Arguments

formula	Symbolic description of the model to be fit.
type	List specifying the type of spline for each variable. See mkterm for details.
data	Optional data frame containing the variables in the model.
weights	Optional vector of weights to be used in the fitting process.
subset	Optional vector specifying a subset of observations to be used in the fitting process.
offset	Optional offset term with known parameter 1.
na.action	Function which indicates what should happen when the data contain NAs.
partial	Optional extra unpenalized terms in partial spline models.
method	Method for smoothing parameter selection. Supported are <code>method="v"</code> for GCV, <code>method="m"</code> for GML (REML), and <code>method="u"</code> for Mallows' CL.
alpha	Parameter modifying GCV or Mallows' CL; larger absolute values yield smoother fits; negative value invokes a stable and more accurate GCV/CL evaluation algorithm but may take two to five times as long. Ignored when <code>method="m"</code> are specified.
varht	External variance estimate needed for <code>method="u"</code> . Ignored when <code>method="v"</code> or <code>method="m"</code> are specified.
id.basis	Index designating selected "knots".
nbasis	Number of "knots" to be selected. Ignored when <code>id.basis</code> is supplied.
seed	Seed to be used for the random generation of "knots". Ignored when <code>id.basis</code> is supplied.
random	Input for parametric random effects in nonparametric mixed-effect models. See mkran for details.

Details

The model specification via `formula` is intuitive. For example, `y~x1*x2` yields a model of the form

$$y = C + f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2) + e$$

with the terms denoted by "1", "x1", "x2", and "x1:x2".

The model terms are sums of unpenalized and penalized terms. Attached to every penalized term there is a smoothing parameter, and the model complexity is largely determined by the number of smoothing parameters.

A subset of the observations are selected as "knots." Unless specified via `id.basis` or `nbasis`, the number of "knots" q is determined by $\max(30, 10n^{2/9})$, which is appropriate for the default cubic splines for numerical vectors.

Using q "knots," `ssanova` calculates an approximate solution to the penalized least squares problem using algorithms of the order $O(nq^2)$, which for $q \ll n$ scale better than the $O(n^3)$ algorithms of `ssanova0`. For the exact solution, one may set $q = n$ in `ssanova`, but `ssanova0` would be much faster.

Value

`ssanova` returns a list object of class "ssanova".

The method `summary.ssanova` can be used to obtain summaries of the fits. The method `predict.ssanova` can be used to evaluate the fits at arbitrary points along with standard errors. The method `project.ssanova` can be used to calculate the Kullback-Leibler projection for model selection. The methods `residuals.ssanova` and `fitted.ssanova` extract the respective traits from the fits.

Note

To use GCV and Mallows' CL unmodified, set `alpha=1`.

For simpler models and moderate sample sizes, the exact solution of `ssanova0` can be faster.

The results may vary from run to run. For consistency, specify `id.basis` or set `seed`.

In `gss` versions earlier than 1.0, `ssanova` was under the name `ssanova1`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

- Gu, C. (2002), *Smoothing Spline ANOVA Models*. New York: Springer-Verlag.
- Kim, Y.-J. and Gu, C. (2004), Smoothing spline Gaussian regression: more scalable computation via efficient approximation. *Journal of the Royal Statistical Society, Ser. B*, **66**, 337–356.
- Wahba, G. (1990), *Spline Models for Observational Data*. Philadelphia: SIAM.

Examples

```
## Fit a cubic spline
x <- runif(100); y <- 5 + 3*sin(2*pi*x) + rnorm(x)
cubic.fit <- ssanova(y~x)
## Obtain estimates and standard errors on a grid
new <- data.frame(x=seq(min(x),max(x),len=50))
est <- predict(cubic.fit,new,se=TRUE)
## Plot the fit and the Bayesian confidence intervals
plot(x,y,col=1); lines(new$x,est$fit,col=2)
lines(new$x,est$fit+1.96*est$se,col=3)
lines(new$x,est$fit-1.96*est$se,col=3)
## Clean up
## Not run:
rm(x,y,cubic.fit,new,est)
dev.off()
## End(Not run)

## Fit a tensor product cubic spline
data(nox)
nox.fit <- ssanova(log10(nox)~comp*equi,data=nox)
## Fit a spline with cubic and nominal marginals
nox$comp<-as.factor(nox$comp)
nox.fit.n <- ssanova(log10(nox)~comp*equi,data=nox)
## Fit a spline with cubic and ordinal marginals
nox$comp<-as.ordered(nox$comp)
nox.fit.o <- ssanova(log10(nox)~comp*equi,data=nox)
## Clean up
## Not run: rm(nox,nox.fit,nox.fit.n,nox.fit.o)
```

ssanova0

Fitting Smoothing Spline ANOVA Models

Description

Fit smoothing spline ANOVA models in Gaussian regression. The symbolic model specification via formula follows the same rules as in [lm](#).

Usage

```
ssanova0(formula, type=NULL, data=list(), weights, subset,
         offset, na.action=na.omit, partial=NULL, method="v",
         varht=1, prec=1e-7, maxiter=30)
```

Arguments

formula	Symbolic description of the model to be fit.
type	List specifying the type of spline for each variable. See mkterm for details.
data	Optional data frame containing the variables in the model.

<code>weights</code>	Optional vector of weights to be used in the fitting process.
<code>subset</code>	Optional vector specifying a subset of observations to be used in the fitting process.
<code>offset</code>	Optional offset term with known parameter 1.
<code>na.action</code>	Function which indicates what should happen when the data contain NAs.
<code>partial</code>	Optional extra unpenalized terms in partial spline models.
<code>method</code>	Method for smoothing parameter selection. Supported are <code>method="v"</code> for GCV, <code>method="m"</code> for GML (REML), and <code>method="u"</code> for Mallow's CL.
<code>varht</code>	External variance estimate needed for <code>method="u"</code> . Ignored when <code>method="v"</code> or <code>method="m"</code> are specified.
<code>prec</code>	Precision requirement in the iteration for multiple smoothing parameter selection. Ignored when only one smoothing parameter is involved.
<code>maxiter</code>	Maximum number of iterations allowed for multiple smoothing parameter selection. Ignored when only one smoothing parameter is involved.

Details

The model specification via `formula` is intuitive. For example, `y~x1*x2` yields a model of the form

$$y = C + f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2) + e$$

with the terms denoted by "1", "x1", "x2", and "x1:x2".

The model terms are sums of unpenalized and penalized terms. Attached to every penalized term there is a smoothing parameter, and the model complexity is largely determined by the number of smoothing parameters.

`ssanova0` and the affiliated methods provide a front end to RKPACk, a collection of RATFOR routines for nonparametric regression via the penalized least squares. The algorithms implemented in RKPACk are of the order $O(n^3)$.

Value

`ssanova0` returns a list object of class `c("ssanova0", "ssanova")`.

The method `summary.ssanova0` can be used to obtain summaries of the fits. The method `predict.ssanova0` can be used to evaluate the fits at arbitrary points along with standard errors. The methods `residuals.ssanova` and `fitted.ssanova` extract the respective traits from the fits.

Note

For complex models and large sample sizes, the approximate solution of `ssanova` can be faster.

The method `project` is not implemented for `ssanova0`, nor is the mixed-effect model support through `mkran`.

In `gss` versions earlier than 1.0, `ssanova0` was under the name `ssanova`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

- Gu, C. (2002), *Smoothing Spline ANOVA Models*. New York: Springer-Verlag.
 Wahba, G. (1990), *Spline Models for Observational Data*. Philadelphia: SIAM.

Examples

```
## Fit a cubic spline
x <- runif(100); y <- 5 + 3*sin(2*pi*x) + rnorm(x)
cubic.fit <- ssanova0(y~x,method="m")
## Obtain estimates and standard errors on a grid
new <- data.frame(x=seq(min(x),max(x),len=50))
est <- predict(cubic.fit,new,se=TRUE)
## Plot the fit and the Bayesian confidence intervals
plot(x,y,col=1); lines(new$x,est$fit,col=2)
lines(new$x,est$fit+1.96*est$se,col=3)
lines(new$x,est$fit-1.96*est$se,col=3)
## Clean up
## Not run:
rm(x,y,cubic.fit,new,est)
dev.off()
## End(Not run)

## Fit a tensor product cubic spline
data(nox)
nox.fit <- ssanova0(log10(nox)~comp*equi,data=nox)
## Fit a spline with cubic and nominal marginals
nox$comp<-as.factor(nox$comp)
nox.fit.n <- ssanova0(log10(nox)~comp*equi,data=nox)
## Fit a spline with cubic and ordinal marginals
nox$comp<-as.ordered(nox$comp)
nox.fit.o <- ssanova0(log10(nox)~comp*equi,data=nox)
## Clean up
## Not run: rm(nox,nox.fit,nox.fit.n,nox.fit.o)
```

Description

Estimate probability densities using smoothing spline ANOVA models. The symbolic model specification via `formula` follows the same rules as in [lm](#), but with the response missing.

Usage

```
ssden(formula, type=NULL, data=list(), alpha=1.4, weights=NULL,
      subset, na.action=na.omit, id.basis=NULL, nbasis=NULL, seed=NULL,
      domain=as.list(NULL), quadrature=NULL, prec=1e-7, maxiter=30)
```

Arguments

<code>formula</code>	Symbolic description of the model to be fit.
<code>type</code>	List specifying the type of spline for each variable. See mkterm for details.
<code>data</code>	Optional data frame containing the variables in the model.
<code>alpha</code>	Parameter defining cross-validation score for smoothing parameter selection.
<code>weights</code>	Optional vector of bin-counts for histogram data.
<code>subset</code>	Optional vector specifying a subset of observations to be used in the fitting process.
<code>na.action</code>	Function which indicates what should happen when the data contain NAs.
<code>id.basis</code>	Index of observations to be used as "knots."
<code>nbasis</code>	Number of "knots" to be used. Ignored when <code>id.basis</code> is specified.
<code>seed</code>	Seed to be used for the random generation of "knots." Ignored when <code>id.basis</code> is specified.
<code>domain</code>	Data frame specifying marginal support of density.
<code>quadrature</code>	Quadrature for calculating integral. Mandatory if variables other than factors or numerical vectors are involved.
<code>prec</code>	Precision requirement for internal iterations.
<code>maxiter</code>	Maximum number of iterations allowed for internal iterations.

Details

The model specification via `formula` is for the log density. For example, `~x1*x2` prescribes a model of the form

$$\log f(x_1, x_2) = g_1(x_1) + g_2(x_2) + g_{12}(x_1, x_2) + C$$

with the terms denoted by "`x1`", "`x2`", and "`x1 : x2`"; the constant is determined by the fact that a density integrates to one.

The selective term elimination may characterize (conditional) independence structures between variables. For example, `~x1*x2+x1*x3` yields the conditional independence of `x2` and `x3` given `x1`.

Parallel to those in a [ssanova](#) object, the model terms are sums of unpenalized and penalized terms. Attached to every penalized term there is a smoothing parameter, and the model complexity is largely determined by the number of smoothing parameters.

The selection of smoothing parameters is through a cross-validation mechanism described in the references, with a parameter `alpha`; `alpha=1` is "unbiased" for the minimization of Kullback-Leibler loss but may yield severe undersmoothing, whereas larger `alpha` yields smoother estimates.

A subset of the observations are selected as "knots." Unless specified via `id.basis` or `nbasis`, the number of "knots" q is determined by $\max(30, 10n^{2/9})$, which is appropriate for the default cubic splines for numerical vectors.

Value

`ssden` returns a list object of class "ssden".

`dssden` and `cdssden` can be used to evaluate the estimated joint density and conditional density; `pssden`, `qssden`, `cpssden`, and `cqssden` can be used to evaluate (conditional) cdf and quantiles. The method `project.ssden` can be used to calculate the Kullback-Leibler projection for model selection.

Note

Default quadrature will be constructed for up to 4 numerical vectors on a hyper cube, then outer product with factor levels will be taken if factors are involved. The sides of the hyper cube are specified by `domain`; for `domain$x` missing, the default is $c(\min(x), \max(x)) + c(-1, 1) * (\max(x) - \min(x)) * .05$.

On a 1-D interval, the quadrature is the 200-point Gauss-Legendre formula returned from `gauss.quad`. For 2, 3, or 4 numerical vectors, delayed Smolyak cubatures from `smolyak.quad` with 449, 2527, and 13697 points are used on cubes with the marginals properly transformed; see Gu and Wang (2003) for the marginal transformations.

The results may vary from run to run. For consistency, specify `id.basis` or set `seed`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (2002), *Smoothing Spline ANOVA Models*. New York: Springer-Verlag.

Gu, C. and Wang, J. (2003), Penalized likelihood density estimation: Direct cross-validation and scalable approximation. *Statistica Sinica*, **13**, 811–826.

Examples

```
## 1-D estimate: Buffalo snowfall
data(buffalo)
buff.fit <- ssden(~buffalo, domain=data.frame(buffalo=c(0, 150)))
plot(xx<-seq(0, 150, len=101), dssden(buff.fit, xx), type="l")
plot(xx, pssden(buff.fit, xx), type="l")
plot(qq<-seq(0, 1, len=51), qssden(buff.fit, qq), type="l")
## Clean up
## Not run:
rm(buffalo, buff.fit, xx, qq)
dev.off()
## End(Not run)

## 2-D with triangular domain: AIDS incubation
```

```

data(aids)
## rectangular quadrature
quad.pt <- expand.grid(incu=((1:40)-.5)/40*100, infe=((1:40)-.5)/40*100)
quad.pt <- quad.pt[quad.pt$incu<=quad.pt$infe,]
quad.wt <- rep(1,nrow(quad.pt))
quad.wt[quad.pt$incu==quad.pt$infe] <- .5
quad.wt <- quad.wt/sum(quad.wt)*5e3
## additive model (pre-truncation independence)
aids.fit <- ssden(~incu+infe,data=aids,subset=age>=60,
                 domain=data.frame(incu=c(0,100), infe=c(0,100)),
                 quad=list(pt=quad.pt, wt=quad.wt))
## conditional (marginal) density of infe
jk <- cdssden(aids.fit, xx<-seq(0,100, len=51), data.frame(incu=50))
plot(xx, jk$pdf, type="l")
## conditional (marginal) quantiles of infe (TIME-CONSUMING)
## Not run:
cqssden(aids.fit, c(.05, .25, .5, .75, .95), data.frame(incu=50), jk$int)
## End(Not run)
## Clean up
## Not run:
rm(aids, quad.pt, quad.wt, aids.fit, jk, xx)
dev.off()
## End(Not run)

## One factor plus one vector
data(gastric)
gastric$trt
fit <- ssden(~futime*trt, data=gastric)
## conditional density
cdssden(fit, c("1", "2"), cond=data.frame(futime=150))
## conditional quantiles
cqssden(fit, c(.05, .25, .5, .75, .95), data.frame(trt="1"))
## Clean up
## Not run: rm(gastric, fit)

```

sshzd

Estimating Hazard Function Using Smoothing Splines

Description

Estimate hazard function using smoothing spline ANOVA models. The symbolic model specification via `formula` follows the same rules as in [lm](#), but with the response of a special form.

Usage

```

sshzd(formula, type=NULL, data=list(), alpha=1.4, weights=NULL,
       subset, na.action=na.omit, id.basis=NULL, nbasis=NULL, seed=NULL,
       prec=1e-7, maxiter=30)

```

Arguments

<code>formula</code>	Symbolic description of the model to be fit, where the response is of the form <code>Surv(futime, status, start=0)</code> .
<code>type</code>	List specifying the type of spline for each variable. See <code>mkterm</code> for details.
<code>data</code>	Optional data frame containing the variables in the model.
<code>alpha</code>	Parameter defining cross-validation score for smoothing parameter selection.
<code>weights</code>	Optional vector of counts for duplicated data.
<code>subset</code>	Optional vector specifying a subset of observations to be used in the fitting process.
<code>na.action</code>	Function which indicates what should happen when the data contain NAs.
<code>id.basis</code>	Index of observations to be used as "knots."
<code>nbasis</code>	Number of "knots" to be used. Ignored when <code>id.basis</code> is specified.
<code>seed</code>	Seed to be used for the random generation of "knots." Ignored when <code>id.basis</code> is specified.
<code>prec</code>	Precision requirement for internal iterations.
<code>maxiter</code>	Maximum number of iterations allowed for internal iterations.

Details

The model specification via `formula` is for the log hazard. For example, `Surv(t, d) ~ t*u` prescribes a model of the form

$$\log f(t, u) = C + g_t(t) + g_u(u) + g_{t,u}(t, u)$$

with the terms denoted by "1", "t", "u", and "t:u". Replacing `t*u` by `t+u` in the formula, one gets a proportional hazard model with $g_{t,u} = 0$.

`sshzd` takes standard right-censored lifetime data, with possible left-truncation and covariates; in `Surv(futime, status, start=0) ~ . . .`, `futime` is the follow-up time, `status` is the censoring indicator, and `start` is the optional left-truncation time. The main effect of `futime` must appear in the model terms specified via . . .

Parallel to those in a `ssanova` object, the model terms are sums of unpenalized and penalized terms. Attached to every penalized term there is a smoothing parameter, and the model complexity is largely determined by the number of smoothing parameters.

The selection of smoothing parameters is through a cross-validation mechanism described in Gu (2002, Sec. 7.2), with a parameter `alpha`; `alpha=1` is "unbiased" for the minimization of Kullback-Leibler loss but may yield severe undersmoothing, whereas larger `alpha` yields smoother estimates.

A subset of the observations are selected as "knots." Unless specified via `id.basis` or `nbasis`, the number of "knots" q is determined by $\max(30, 10n^{2/9})$, which is appropriate for the default cubic splines for numerical vectors.

Value

sshzd returns a list object of `class` "sshzd".

`hzdrate.sshzd` can be used to evaluate the estimated hazard function. `hzdcurve.sshzd` can be used to evaluate hazard curves with fixed covariates. `survexp.sshzd` can be used to calculate estimated expected survival. The method `project.sshzd` can be used to calculate the Kullback-Leibler projection for model selection.

Note

The function `Surv(futime, status, start=0)` is defined and parsed inside sshzd, not quite the same as the one in the `survival` package.

Integration on the time axis is done by the 200-point Gauss-Legendre formula on `c(min(start), max(futime))`, returned from `gauss.quad`.

The results may vary from run to run. For consistency, specify `id.basis` or `set.seed`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (2002), *Smoothing Spline ANOVA Models*. New York: Springer-Verlag.

Du, P. and Gu, C. (2006), Penalized likelihood hazard estimation: efficient approximation and Bayesian confidence intervals. *Statistics and Probability Letters*, **76**, 244–254.

Examples

```
## Model with interaction
data(gastric)
gastric.fit <- sshzd(Surv(futime, status)~futime*trt, data=gastric)
## exp(-Lambda(600)), exp(-(Lambda(1200)-Lambda(600))), and exp(-Lambda(1200))
survexp.sshzd(gastric.fit, c(600, 1200, 1200), data.frame(trt=as.factor(1)), c(0, 600, 0))
## Clean up
## Not run:
rm(gastric, gastric.fit)
dev.off()
## End(Not run)

## THE FOLLOWING EXAMPLE IS TIME-CONSUMING
## Proportional hazard model
## Not run:
data(stan)
stan.fit <- sshzd(Surv(futime, status)~futime+age, data=stan)
## Evaluate fitted hazard
hzdrate.sshzd(stan.fit, data.frame(futime=c(10, 20), age=c(20, 30)))
## Plot lambda(t, age=20)
tt <- seq(0, 60, leng=101)
hh <- hzdcurve.sshzd(stan.fit, tt, data.frame(age=20))
plot(tt, hh, type="l")
```

```
## Clean up
rm(stan, stan.fit, tt, hh)
dev.off()
## End(Not run)
```

stan	<i>Stanford Heart Transplant Data</i>
------	---------------------------------------

Description

Survival of patients from the Stanford heart transplant program.

Usage

```
data(stan)
```

Format

A data frame containing 184 observations on the following variables.

time	Follow-up time after transplant, in days.
status	Censoring status.
age	Age at transplant.
futime	Square root of time.

Source

Miller, R. G. and Halpern, J. (1982), Regression with censored data. *Biometrika*, **69**, 521–531.

summary.gssanova	<i>Assessing Smoothing Spline ANOVA Fits with Non-Gaussian Responses</i>
------------------	--

Description

Calculate various summaries of smoothing spline ANOVA fits with non-Gaussian responses.

Usage

```
## S3 method for class 'gssanova':
summary(object, diagnostics=FALSE, ...)
```

Arguments

object	Object of class "gssanova".
diagnostics	Flag indicating if diagnostics are required.
...	Ignored.

Details

Similar to the iterated weighted least squares fitting of `glm`, penalized likelihood regression fit can be calculated through iterated penalized weighted least squares.

The diagnostics are based on the "pseudo" Gaussian response model behind the weighted least squares problem at convergence.

Value

`summary.gssanova` returns a list object of class `"summary.gssanova"` consisting of the following components. The entries `pi`, `kappa`, `cosines`, and `roughness` are only calculated if `diagnostics=TRUE`.

<code>call</code>	Fitting call.
<code>family</code>	Error distribution.
<code>alpha</code>	Parameter used to define cross-validation in model fitting.
<code>fitted</code>	Fitted values on the link scale.
<code>dispersion</code>	Assumed or estimated dispersion parameter.
<code>residuals</code>	Working residuals on the link scale.
<code>rss</code>	Residual sum of squares.
<code>dev.resid</code>	Deviance residuals.
<code>deviance</code>	Deviance of the fit.
<code>dev.null</code>	Deviance of the null model.
<code>penalty</code>	Roughness penalty associated with the fit.
<code>pi</code>	"Percentage decomposition" of "explained variance" into model terms.
<code>kappa</code>	Concurvity diagnostics for model terms. Virtually the square roots of variance inflation factors of a retrospective linear model.
<code>cosines</code>	Cosine diagnostics for practical significance of model terms.
<code>roughness</code>	Percentage decomposition of the roughness penalty <code>penalty</code> into model terms.

Author(s)

Chong Gu, [⟨chong@stat.purdue.edu⟩](mailto:chong@stat.purdue.edu)

References

Gu, C. (1992), Diagnostics for nonparametric regression models with additive terms. *Journal of the American Statistical Association*, **87**, 1051–1058.

See Also

Fitting function `gssanova` and methods `predict.ssanova`, `project.gssanova`, `fitted.gssanova`.

summary.gssanova0 *Assessing Smoothing Spline ANOVA Fits with Non-Gaussian Responses*

Description

Calculate various summaries of smoothing spline ANOVA fits with non-Gaussian responses.

Usage

```
## S3 method for class 'gssanova0':
summary(object, diagnostics=FALSE, ...)
```

Arguments

object	Object of class "gssanova".
diagnostics	Flag indicating if diagnostics are required.
...	Ignored.

Details

Similar to the iterated weighted least squares fitting of [glm](#), penalized likelihood regression fit can be calculated through iterated penalized weighted least squares.

The diagnostics are based on the "pseudo" Gaussian response model behind the weighted least squares problem at convergence.

Value

summary.gssanova0 returns a list object of `class` "summary.gssanova0" consisting of the following components. The entries `pi`, `kappa`, `cosines`, and `roughness` are only calculated if `diagnostics=TRUE`.

call	Fitting call.
family	Error distribution.
method	Method for smoothing parameter selection.
dispersion	Assumed or estimated dispersion parameter.
iter	Number of performance-oriented iterations performed.
fitted	Fitted values on the link scale.
residuals	Working residuals on the link scale.
rss	Residual sum of squares.
dev.resid	Deviance residuals.
deviance	Deviance of the fit.
dev.null	Deviance of the null model.

alpha	Estimated size for family="nbinomial" with one column responses. Estimated inverse scale of log life time for family="nbinomial", "lognorm", or "loglogis".
penalty	Roughness penalty associated with the fit.
pi	"Percentage decomposition" of "explained variance" into model terms.
kappa	Concurvity diagnostics for model terms. Virtually the square roots of variance inflation factors of a retrospective linear model.
cosines	Cosine diagnostics for practical significance of model terms.
roughness	Percentage decomposition of the roughness penalty <code>penalty</code> into model terms.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (1992), Diagnostics for nonparametric regression models with additive terms. *Journal of the American Statistical Association*, **87**, 1051–1058.

See Also

Fitting function `gssanova0` and methods `predict.ssanova0`, `fitted.gssanova`.

summary.ssanova *Assessing Smoothing Spline ANOVA Fits*

Description

Calculate various summaries of smoothing spline ANOVA fits.

Usage

```
## S3 method for class 'ssanova':
summary(object, diagnostics=FALSE, ...)
## S3 method for class 'ssanova0':
summary(object, diagnostics=FALSE, ...)
```

Arguments

object	Object of class "ssanova".
diagnostics	Flag indicating if diagnostics are required.
...	Ignored.

Value

`summary.ssanova` returns a list object of class "summary.ssanova" consisting of the following components. The entries `pi`, `kappa`, `cosines`, and `roughness` are only calculated if `diagnostics=TRUE`; see the reference below for details concerning the diagnostics.

<code>call</code>	Fitting call.
<code>method</code>	Method for smoothing parameter selection.
<code>fitted</code>	Fitted values.
<code>residuals</code>	Residuals.
<code>sigma</code>	Assumed or estimated error standard deviation.
<code>r.squared</code>	Fraction of "explained variance" by the fitted model.
<code>rss</code>	Residual sum of squares.
<code>penalty</code>	Roughness penalty associated with the fit.
<code>pi</code>	"Percentage decomposition" of "explained variance" into model terms.
<code>kappa</code>	Concurvity diagnostics for model terms. Virtually the square roots of variance inflation factors of a retrospective linear model.
<code>cosines</code>	Cosine diagnostics for practical significance of model terms.
<code>roughness</code>	Percentage decomposition of the roughness penalty <code>penalty</code> into model terms.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (1992), Diagnostics for nonparametric regression models with additive terms. *Journal of the American Statistical Association*, **87**, 1051–1058.

See Also

Fitting functions `ssanova`, `ssanova0` and methods `predict.ssanova`, `project.ssanova`, `fitted.ssanova`.

wesdr

Progression of Diabetic Retinopathy

Description

Data derived from the Wisconsin Epidemiological Study of Diabetic Retinopathy.

Usage

`data(wesdr)`

Format

A data frame containing 669 observations on the following variables.

dur Duration of diabetes at baseline, in years.
gly Percent of glycosylated hemoglobin at baseline.
bmi Body mass index at baseline.
ret Binary indicator of retinopathy progression at first follow-up.

Source

Wang, Y. (1997), GRKPACK: Fitting smoothing spline ANOVA models for exponential families. *Communications in Statistics – Simulations and Computation*, **26**, 765–782.

References

Klein, R., Klein, B. E. K., Moss, S. E., Davis, M. D., and DeMets, D. L. (1988), Glycosylated hemoglobin predicts the incidence and progression of diabetic retinopathy. *Journal of the American Medical Association*, **260**, 2864–2871.

Klein, R., Klein, B. E. K., Moss, S. E., Davis, M. D., and DeMets, D. L. (1989), The Wisconsin Epidemiologic Study of Diabetic Retinopathy. X. Four incidence and progression of diabetic retinopathy when age at diagnosis is 30 or more years. *Archive Ophthalmology*, **107**, 244–249.

Wahba, G., Wang, Y., Gu, C., Klein, R., and Klein, B. E. K. (1995), Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy. *The Annals of Statistics*, **23**, 1865–1895.

Index

*Topic **datasets**

- aids, 1
- bacteriuria, 2
- buffalo, 2
- gastric, 6
- LakeAcidity, 13
- nox, 15
- ozone, 16
- stan, 31
- wesdr, 36

*Topic **distribution**

- cdssden, 3
- dssden, 4
- ssden, 26

*Topic **htest**

- project, 19

*Topic **math**

- gauss.quad, 6
- nlm0, 14
- smolyak, 21

*Topic **models**

- cdssden, 3
- dssden, 4
- fitted.ssanova, 5
- gssanova, 7
- gssanova0, 10
- hzdrate.sshzd, 12
- predict.ssanova, 17
- print, 18
- project, 19
- ssanova, 21
- ssanova0, 24
- ssden, 26
- sshzd, 29
- summary.gssanova, 32
- summary.gssanova0, 33
- summary.ssanova, 35

*Topic **regression**

- fitted.ssanova, 5

- gssanova, 7
- gssanova0, 10
- predict.ssanova, 17
- ssanova, 21
- ssanova0, 24
- summary.gssanova, 32
- summary.gssanova0, 33
- summary.ssanova, 35

*Topic **smooth**

- cdssden, 3
- dssden, 4
- fitted.ssanova, 5
- gssanova, 7
- gssanova0, 10
- hzdrate.sshzd, 12
- predict.ssanova, 17
- print, 18
- project, 19
- ssanova, 21
- ssanova0, 24
- ssden, 26
- sshzd, 29
- summary.gssanova, 32
- summary.gssanova0, 33
- summary.ssanova, 35

*Topic **survival**

- hzdrate.sshzd, 12
- sshzd, 29

aids, 1

bacteriuria, 2
buffalo, 2

cdssden, 3, 5, 27
class, 27, 30, 32, 34, 35
cpssden, 27
cpssden (cdssden), 3
cqssden, 27
cqssden (cdssden), 3

- dssden, [4](#), [4](#), [27](#)
- fitted.gssanova, [8](#), [11](#), [33](#), [34](#)
- fitted.gssanova (*fitted.ssanova*), [5](#)
- fitted.ssanova, [5](#), [18](#), [23](#), [25](#), [36](#)
- gastric, [6](#)
- gauss.quad, [6](#), [27](#), [30](#)
- glm, [7](#), [8](#), [10](#), [11](#), [32](#), [33](#)
- gssanova, [7](#), [11](#), [17–20](#), [33](#)
- gssanova0, [8](#), [10](#), [18](#), [19](#), [34](#)
- hzdcurve.sshzd, [30](#)
- hzdcurve.sshzd (*hzdrate.sshzd*), [12](#)
- hzdrate.sshzd, [12](#), [30](#)
- LakeAcidity, [13](#)
- lm, [7](#), [10](#), [21](#), [24](#), [26](#), [29](#)
- mkran, [7](#), [11](#), [22](#), [25](#)
- mkterm, [7](#), [10](#), [22](#), [24](#), [26](#), [29](#)
- NegBinomial, [8](#), [11](#)
- nlm0, [14](#)
- nox, [15](#)
- ozone, [16](#)
- predict.lm, [3](#), [4](#)
- predict.ssanova, [8](#), [17](#), [23](#), [33](#), [36](#)
- predict.ssanova0, [11](#), [25](#), [34](#)
- predict.ssanova0
(*predict.ssanova*), [17](#)
- print, [18](#)
- print.summary.gssanova (*print*), [18](#)
- print.summary.gssanova0 (*print*), [18](#)
- print.summary.ssanova (*print*), [18](#)
- project, [11](#), [19](#), [25](#)
- project.gssanova, [8](#), [33](#)
- project.ssanova, [18](#), [23](#), [36](#)
- project.ssdn, [27](#)
- project.sshzd, [30](#)
- pssden, [27](#)
- pssden (*dssden*), [4](#)
- qssden, [27](#)
- qssden (*dssden*), [4](#)
- residuals.gssanova, [8](#), [11](#)
- residuals.gssanova
(*fitted.ssanova*), [5](#)
- residuals.ssanova, [23](#), [25](#)
- residuals.ssanova
(*fitted.ssanova*), [5](#)
- smolyak, [21](#)
- smolyak.quad, [27](#)
- ssanova, [17–20](#), [21](#), [25](#), [27](#), [30](#), [36](#)
- ssanova0, [18](#), [19](#), [22](#), [23](#), [24](#), [36](#)
- ssden, [4](#), [5](#), [19](#), [20](#), [26](#)
- sshzd, [13](#), [19](#), [20](#), [29](#)
- stan, [31](#)
- summary.gssanova, [8](#), [18](#), [19](#), [32](#)
- summary.gssanova0, [11](#), [18](#), [19](#), [33](#)
- summary.ssanova, [18](#), [19](#), [23](#), [35](#)
- summary.ssanova0, [25](#)
- summary.ssanova0
(*summary.ssanova*), [35](#)
- survexp.sshzd, [30](#)
- survexp.sshzd (*hzdrate.sshzd*), [12](#)
- wesdr, [36](#)