Package ‘cgam’

February 19, 2015

Type Package
Title Constrained Generalized Additive Model
Version 1.2
Date 2014-09-10
Author Mary C. Meyer and Xiyue Liao
Maintainer Xiyue Liao<xiyue@rams.colostate.edu>
Description In this package, a constrained generalized additive model is fitted by the cgam routine. Given a set of predictors with or without shape or order restrictions, the maximum likelihood estimator for the constrained generalized additive model is found using an iteratively re-weighted cone projection algorithm. The cone information criterion (CIC) may be used to select the best combination of variables and shapes. This package depends on the R package cone-proj.
License GPL (>= 2)
Depends coneproj(>= 1.3), R(>= 3.0.2)
NeedsCompilation no
Suggests stats, MASS, graphics, grDevices, utils
URL http://cran.r-project.org/web/packages/cgam
Repository CRAN
Date/Publication 2014-09-10 21:48:13

R topics documented:

  cgam .......................................................... 2
  conc .......................................................... 8
  conv .......................................................... 10
  cubic .......................................................... 11
  decr .......................................................... 12
  decr.conc ..................................................... 13
  decr.conv .................................................... 15
  incr .......................................................... 16
  incr.conc ..................................................... 18
Constrained Generalized Additive Model Fitting

**Description**

The partial linear generalized additive model is fitted using the method of maximum likelihood, where shape or order restrictions can be imposed on the non-parametrically modelled predictors with optional smoothing, and no restrictions are imposed on the optional parametrically modelled categorical covariates.

**Usage**

```r
cgam(formula, nsim = 1e+2, family = gaussian(), data = NULL, weights = NULL)
```

**Arguments**

- `formula`: A formula object which gives a symbolic description of the model to be fitted. It has the form "response ~ predictor". The response is a vector of length n. The specification of the model can be one of the three exponential families: gaussian, binomial and poisson. The systematic component $\eta$ is $E(y)$, the log odds of $y = 1$, and the logarithm of $E(y)$ respectively. A predictor can be a non-parametrically modelled variable with or without a shape or order restriction, or a parametrically modelled unconstrained categorical covariate. In terms of a non-parametrically modelled predictor, the user is supposed to indicate the relationship between the systematic component $\eta$ and a predictor $x$ in the following way:

  Assume that $\eta$ is the systematic component and $x$ is a predictor:

  - `incr(x)`: $\eta$ is increasing in $x$. See `incr` for more details.
  - `s.incr(x)`: $\eta$ is smoothly increasing in $x$. See `s.incr` for more details.
  - `decr(x)`: $\eta$ is decreasing in $x$. See `decr` for more details.
• s.decr(x): \( \eta \) is smoothly decreasing in \( x \). See \texttt{s.decr} for more details.
• conc(x): \( \eta \) is concave in \( x \). See \texttt{conc} for more details.
• s.conc(x): \( \eta \) is smoothly concave in \( x \). See \texttt{sNconc} for more details.
• conv(x): \( \eta \) is convex in \( x \). See \texttt{conv} for more details.
• s.conv(x): \( \eta \) is smoothly convex in \( x \). See \texttt{sNconv} for more details.
• incr.conc(x): \( \eta \) is increasing and concave in \( x \). See \texttt{incr.conc} for more details.
• s.incr.conc(x): \( \eta \) is smoothly increasing and concave in \( x \). See \texttt{sNincrNconc} for more details.
• decr.conc(x): \( \eta \) is decreasing and concave in \( x \). See \texttt{decrNconc} for more details.
• s.decr.conc(x): \( \eta \) is smoothly decreasing and concave in \( x \). See \texttt{sNdecrNconc} for more details.
• incr.conv(x): \( \eta \) is increasing and convex in \( x \). See \texttt{incrNconv} for more details.
• s.incr.conv(x): \( \eta \) is smoothly increasing and convex in \( x \). See \texttt{sNincrNconv} for more details.
• decr.conv(x): \( \eta \) is decreasing and convex in \( x \). See \texttt{decrNconv} for more details.
• s.decr.conv(x): \( \eta \) is smoothly decreasing and convex in \( x \). See \texttt{sNdecrNconv} for more details.
• s(x): \( \eta \) is smooth in \( x \). See \texttt{s} for more details.
• tree(x): \( \eta \) has a tree-ordering in \( x \). See \texttt{tree} for more details.
• umbrella(x): \( \eta \) has an umbrella-ordering in \( x \). See \texttt{umbrella} for more details.

\texttt{nsim}  
The number of simulations used to get the cic parameter. Note that when there is no shape-restricted or order-restricted predictor, \texttt{nsim} has the default 0. Otherwise, the default is \texttt{nsim = 1e+2}.

\texttt{family}  
A parameter indicating the error distribution and link function to be used in the model. It can be a character string naming a family function or the result of a call to a family function. This is borrowed from the \texttt{glm} routine in the \texttt{stats} package. There are three families used in \texttt{cgam}: gaussian, binomial and poisson.

\texttt{data}  
An optional data frame, list or environment containing the variables in the model. The default is \texttt{data = NULL}.

\texttt{weights}  
An optional non-negative vector of "replicate weights" which has the same length as the response vector. If weights are not given, all weights are taken to equal 1. The default is \texttt{weights = NULL}.

\textbf{Details}  
We consider generalized partial linear models with independent observations from an exponential family of the form
\[ p(y_i; \theta, \tau) = \exp\{y_i\theta_i - b(\theta_i)\} \tau - c(y_i, \tau), \quad i = 1, \ldots, n, \]
where the specifications of the functions \( b \) and \( c \) determine the sub-family of models. The mean vector \( \mu = E(y) \) has values \( \mu_i = b'(\theta_i) \), and is related to a design matrix of predictor variables through a monotonically increasing link function \( g(\mu_i) = \eta_i, \quad i = 1, \ldots, n, \) where \( \eta \) is the systematic component and...
describes the relationship with the predictors. The relationship between \( \eta \) and \( \theta \) is determined by the link function \( b \).

For the additive model, the systematic component is specified for each observation by

\[
\eta_i = f_1(x_{1i}) + \ldots + f_L(x_{Li}) + \beta' z_i,
\]

where the functions \( f_l \) describe the relationships of the non-parametrically modelled predictors \( x_{lj} \), \( \beta \) is a parameter vector, and \( z_i \) contains the values of variables to be modelled parametrically. The non-parametric components are modelled with shape or order assumptions with optional smoothing, and the solution is obtained through an iteratively re-weighted cone projection, with no back-fitting of individual components.

Suppose that \( \eta \) is a \( n \times 1 \) vector. The matrix form of the systematic component and the predictor is

\[
\eta = \phi_1 + \ldots + \phi_L + Z\beta,
\]

where \( \phi_l \) is the individual component for the \( l \)th non-parametrically modelled predictor, \( l = 1,\ldots,L \), and \( Z \) is an \( n \times p \) design matrix for the parametrically modelled predictors.

The constraints for the component \( \phi_l \) are in \( C_l \), where

\[
C_l = \{ \phi : A_l \phi \geq 0 \text{ and } B_l \phi = 0 \}, \quad \text{for matrices } A_l \text{ and } B_l.
\]

The set \( C_l \) is a convex cone and the set \( C = C_1 + \ldots + C_L + Z \) is also a convex cone with a finite set of edges, where the edges are the generators of \( C \), and \( Z \) is the column space of the design matrix \( Z \) for the parametrically modelled predictors.

An iteratively re-weighted cone projection algorithm is used to fit the generalized regression model over the cone \( C \).

See references cited in this section and the official manual (http://cran.r-project.org/web/packages/coneproj/index.html) for the R package coneproj for more details.

**Value**

- \( \mathbf{vhat} \): The fitted value of the systematic component \( \eta \) on the null space of the constraint set.
- \( \mathbf{etahat} \): The fitted systematic component \( \eta \).
- \( \mathbf{muhat} \): The fitted mean value, obtained by transforming the systematic component \( \eta \) by the inverse of the link function.
- \( \mathbf{vcoefs} \): The estimated coefficients for the basis spanning the null space of the constraint set.
- \( \mathbf{xcoefs} \): The estimated coefficients for the edges corresponding to the smooth predictors with no shape constraint and shape-restricted predictors.
- \( \mathbf{zcoefs} \): The estimated coefficients for the parametrically modelled categorical covariates, i.e., the estimation for the vector \( \beta \).
- \( \mathbf{ucoefs} \): The estimated coefficients for the edges corresponding to the predictors with an umbrella-ordering constraint.
- \( \mathbf{tcoefs} \): The estimated coefficients for the edges corresponding to the predictors with a tree-ordering constraint.
- \( \mathbf{coefs} \): The estimated coefficients for the basis spanning the null space of the constraint set and edges corresponding to the shape-restricted and order-restricted predictors.

\( \mathbf{cic} \): The cone information criterion proposed in Meyer(2013a). It uses the "null expected degrees of freedom" as a measure of the complexity of the model. See Meyer(2013a) for further details of cic.
d0 The dimension of the null space contained in the cone generated by all constraint conditions.
edf0 The estimated "null expected degrees of freedom". It is a measure of the complexity of the model. See Meyer (2013a) and Meyer (2013b) for further details.
etacoms The fitted systematic component value for non-parametrically modelled predictors. It is a matrix of which each row is the fitted systematic component value for a non-parametrically modelled predictor. If there are more than one such predictors, the order of the rows is the same as the order that the user defines such predictors in the formula argument of cgam.
xmat A matrix whose columns represent the shape-restricted predictors and smooth predictors with no shape constraint.
zmat A matrix whose columns represent the parametrically modelled categorical covariates. The user can choose to include a constant vector in it or not. It must be of full column rank.
tr A matrix whose columns represent the predictors with a tree-ordering constraint.
umb A matrix whose columns represent the predictors with an umbrella-ordering constraint.
tree.delta A matrix whose rows are the edges corresponding to the predictors with a tree-ordering constraint.
umbrella.delta A matrix whose rows are the edges corresponding to the predictors with an umbrella-ordering constraint.
bigmat A matrix whose rows are the basis spanning the null space of the constraint set and the edges corresponding to the shape-restricted and order-restricted predictors.
shapes A vector including the shape constraints in a cgam fit.
wts The weights in the final iteration of the iteratively re-weighted cone projections.
wts.iter A logical scalar indicating if or not iteratively re-weighted cone projections are used to get the fit. If the error distribution is gaussian, then wt.iter = FALSE; if the error distribution is binomial or poisson, then wt.iter = TRUE.
family The family parameter defined in a cgam formula.
SSE0 The sum of squared residuals for the linear part.
SSE1 The sum of squared residuals for the full model.
pvals.beta The approximate p-values for the estimation of the vector $\beta$. A t-distribution is used as the approximate distribution.
se.beta The standard errors for the estimation of the vector $\beta$.
null.df The degree of freedom for the null model of a cgam fit, i.e., the model only containing a constant vector.
null_deviance The deviance for the null model of a cgam fit, i.e., the model only containing a constant vector.
deviance The residual deviance of a cgam fit.
The terms objects extracted by the generic function `terms` from a cgam fit. See the official help page (http://stat.ethz.ch/R-manual/R-patched/library/stats/html/terms.html) of the `terms` function for more details.

capm
The number of edges corresponding to the shape-restricted predictors.

capms
The number of edges corresponding to the smooth predictors with no shape constraint.

capk
The number of non-constant columns of `zmat`.

capt
The number of edges corresponding to the tree-ordering predictors.

capu
The number of edges corresponding to the umbrella-ordering predictors.

xid1
A vector keeping track of the beginning position of the set of edges in bigmat for each shape-restricted predictor and smooth predictor with no shape constraint in `xmat`.

xid2
A vector keeping track of the end position of the set of edges in bigmat for each shape-restricted predictor and smooth predictor with no shape constraint in `xmat`.

tid1
A vector keeping track of the beginning position of the set of edges in bigmat for each tree-ordering factor in `tr`.

tid2
A vector keeping track of the end position of the set of edges in bigmat for each tree-ordering factor in `tr`.

uid1
A vector keeping track of the beginning position of the set of edges in bigmat for each umbrella-ordering factor in `umb`.

uid2
A vector keeping track of the end position of the set of edges in bigmat for each umbrella-ordering factor in `umb`.

zid
A vector keeping track of the positions of categorical variables in a cgam formula.

vals
A vector storing the smallest value of each categorical variable defined as a factor(matrix).

zid1
A vector keeping track of the beginning position of the levels(columns) in bigmat for each categorical variable defined as a factor(matrix).

zid2
A vector keeping track of the end position of the levels(columns) in bigmat for each categorical variable defined as a factor(matrix).

nsim
The number of simulations used to get the cic parameter.

xmatnms
A vector storing the names of the shape-restricted predictors and the smooth predictors with no shape constraint in `xmat`.

ynm
The name of the response variable.

znms
A vector storing the names of the categorical variables.

is_fac
A logical scalar showing if a categorical variable is a factor(matrix) or not.

knots
A list storing the knots used for each shape-restricted predictor and smooth predictor with no shape constraint. For a smooth, constrained and a smooth, unconstrained predictor, `knots` is a vector of more than 1 elements, and for a shape-restricted predictor without smoothing, `knots = 0`. 
numknots  A vector storing the number of knots for each shape-restricted predictor and smooth predictor with no shape constraint. For a smooth, constrained and a smooth, unconstrained predictor, numknots > 1, and for a shape-restricted predictor without smoothing, numknots = 0.

sps  A character vector storing the space parameter to create knots for each shape-restricted predictor.

call  The matched call.

ms  The centering terms used to make the C-spline or I-spline edges for shape-restricted predictors with smoothing and smooth, unconstrained predictors. See references cited in this section for more details.

Author(s)
Mary C. Meyer and Xiyue Liao

References

Examples
# Example 1.
data(cubic)
# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x with no restriction with lm()
fit.lm <- lm(y ~ x + I(x^2) + I(x^3))
# regress y on x under the restriction: "increasing and convex"
fit.cgam <- cgam(y ~ incr.conv(x))

# make a plot to compare the two fits
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, fit.cgam$muhat, col = 2, lty = 2)
lines(x, fitted(fit.lm), col = 1, lty = 1)
legend("topleft", bty = "n", c("constrained cgam fit", "unconstrained lm fit"),
       lty = c(2, 1), col = c(2, 1))

# Example 2.
## Not run:
library(gam)
data(kyphosis)

# regress Kyphosis on Age, Number, and Start under the restrictions:
# "concave", "increasing and concave", and "decreasing and concave"
fit <- cgam(Kyphosis ~ conc(Age) + incr.conc(Number) + decr.conc(Start),
            family = binomial(), data = kyphosis)

## End(Not run)

# Example 3.
library(MASS)
data(Rubber)

# regress loss on hard and tens under the restrictions:
# "decreasing" and "decreasing"
fit.cgam <- cgam(loss ~ decr(hard) + decr(tens), data = Rubber)
# "smooth and decreasing" and "smooth and decreasing"
fit.cgam.s <- cgam(loss ~ s.decr(hard) + s.decr(tens), data = Rubber)
hard <- Rubber$hard
tens <- Rubber$tens

# make a 3D plot based on fit.cgam and fit.cgam.s
plotpersp(fit.cgam, hard, tens, th = 120, main = "3D Plot of a Cgam Fit")
plotpersp(fit.cgam.s, hard, tens, th = 120, main = "3D Plot of a Smooth Cgam Fit")

---

conc

Specify a Concave Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the systematic component $\eta$ is concave in a predictor in a formula argument to cgam. This is the unsmoothed version.

Usage

conc(x, numknots = 0, knots = 0, space = "Q")
**conc**

**Arguments**

- \(x\) A numeric predictor which has the same length as the response vector.
- `numknots` The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- `knots` The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- `space` A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "Q".

**Details**

"conc" returns the vector "x" and imposes on it four attributes: name, numknots, knots, and shape.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "conc"; the shape attribute is 4("concave"), and according to the value of the vector itself and its shape attribute, the cone edges of the cone generated by the constraint matrix, which constrains the relationship between the systematic component \(\eta\) and "x" to be concave, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "conc" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

**Value**

The vector x with four attributes, i.e., name: the name of x; shape: 4("concave"); numknots: the numknots argument in "conc"; knots: the knots argument in "conc".

**Author(s)**

Mary C. Meyer and Xiyue Liao

**References**


**See Also**

conv

**Examples**

```r
# generate y
x <- seq(-1, 2, by = 0.1)
n <- length(x)
y <- - x^2 + rnorm(n, .3)

# regress y on x under the shape-restriction: "concave"
```
ans <- cgam(y ~ conc(x))

# make a plot
plot(x, y)
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "concave fit", col = 2, lty = 1)
Author(s)
Mary C. Meyer and Xiyue Liao

References

See Also
conc

Examples

```
# generate y
x <- seq(-1, 2, by = 0.1)
n <- length(x)
y <- x^2 + rnorm(n, .3)

# regress y on x under the shape-restriction: "convex"
ans <- cgam(y ~ conv(x, 3))

# make a plot
plot(x, y)
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "convex fit", col = 2, lty = 1)
```

---

**cubic**  
* A Data Set for Cgam

Description
This data set is used for several examples in the cgam package.

Usage
data(cubic)

Format
A data frame with 50 observations on the following 2 variables.

- **x** The predictor vector.
- **y** The response vector.

Source
STAT640 HW 14 given by Dr. Meyer.
**decr**  
*Specify a Decreasing Shape-Restriction in a CGAM Formula*

### Description

A symbolic routine to define that the systematic component $\eta$ is decreasing in a predictor in a formula argument to cgam. This is the unsmoothed version.

### Usage

```r
decr(x, numknots = 0, knots = 0, space = "Q")
```

### Arguments

- **x**: A numeric predictor which has the same length as the response vector.
- **numknots**: The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- **knots**: The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- **space**: A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "Q".

### Details

"decr" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots. The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "decr"; the shape attribute is 2("decreasing"), and according to the value of the vector itself and its shape attribute, the cone edges of the cone generated by the constraint matrix, which constrains the relationship between the systematic component $\eta$ and "x" to be decreasing, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "decr" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

### Value

The vector x with four attributes, i.e., name: the name of x; shape: 2("decreasing"); numknots: the numknots argument in "decr"; knots: the knots argument in "decr".

### Author(s)

Mary C. Meyer and Xiyue Liao
References


See Also
decr.conc, decr.conv

Examples

data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x with the shape restriction: “decreasing”
ans <- cgam(y ~ decr(x))

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("bottomright", bty = "n", "decreasing fit", col = 2, lty = 1)

Specifying a Decreasing and Concave Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the systematic component $\eta$ is decreasing and concave in a predictor in a formula argument to cgam. This is the unsmoothed version.

Usage
decr.conc(x, numknots = 0, knots = 0, space = "Q")

Arguments

- **x**
  A numeric predictor which has the same length as the response vector.

- **numknots**
  The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.

- **knots**
  The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "Q".

Details

"decr.conc" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "decr.conc"; the shape attribute is 8("decreasing and concave"), and according to the value of the vector itself and its shape attribute, the cone edges of the cone generated by the constraint matrix, which constrains the relationship between the systematic component $\eta$ and "x" to be decreasing and concave, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "decr.conc" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 8("decreasing and concave"); numknots: the numknots argument in "decr.conc"; knots: the knots argument in "decr.conc".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

decl.conv, decl

Examples

data(cubic)

  # extract x
  x <- cubic$x

  # extract y
  y <- cubic$y

  # regress y on x with the shape restriction: "decreasing" and "concave"
  ans <- cgam(y ~ decr.conc(x))

  # make a plot
  par(mar = c(4, 4, 1, 1))
decr.conv

plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "decreasing and concave fit", col = 2, lty = 1)

---

**decr.conv**

*Specify a Decreasing and Convex Shape-Restriction in a CGAM Formula*

**Description**

A symbolic routine to define that the systematic component $\eta$ is decreasing and convex in a predictor in a formula argument to cgam. This is the unsmoothed version.

**Usage**

decr.conv(x, numknots = 0, knots = 0, space = "Q")

**Arguments**

- **x** A numeric predictor which has the same length as the response vector.
- **numknots** The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- **knots** The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- **space** A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "Q".

**Details**

"decr.conv" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "decr.conv"; the shape attribute is 6("decreasing and convex"), and according to the value of the vector itself and its shape attribute, the cone edges of the cone generated by the constraint matrix, which constrains the relationship between the systematic component $\eta$ and "x" to be decreasing and convex, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "decr.conv" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

**Value**

The vector x with four attributes, i.e., name: the name of x; shape: 6("decreasing and convex"); numknots: the numknots argument in "decr.conv"; knots: the knots argument in "decr.conv".
Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also
deccr, conc, decr

Examples

data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x with the shape restriction: "decreasing" and "convex"
ans <- cgam(y ~ decr.conv(x))

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("bottomright", bty = "n", "decreasing and convex fit", col = 2, lty = 1)

---

**incr**

*Specify an Increasing Shape-Restriction in a CGAM Formula*

Description

A symbolic routine to define that the systematic component \( \eta \) is increasing in a predictor in a formula argument to cgam. This is the unsmoothed version.

Usage

\[
icr(x, \text{numknots} = \emptyset, \text{knots} = \emptyset, \text{space} = "Q")\]
Arguments

- **x**: A numeric predictor which has the same length as the response vector.
- **numknots**: The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- **knots**: The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
- **space**: A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "Q".

Details

"incr" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots. The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "incr"; the shape attribute is 1("increasing"), and according to the value of the vector itself and its attributes, the cone edges of the cone generated by the constraint matrix, which constrains the relationship between the systematic component \( \eta \) and "x" to be increasing, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "incr" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 1("increasing"); numknots: the numknots argument in "incr"; knots: the knots argument in "incr".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

- incr.conc, incr.conv

Examples

```r
data(cubic)

# extract x
x <- cubic$x

# extract y
```
y <- cubic$y

# regress y on x with the shape restriction: "increasing"
ans <- cgam(y ~ incr(x))

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "increasing fit", col = 2, lty = 1)

---

**incr.conc**  
*Specify an Increasing and Concave Shape-Restriction in a CGAM Formula*

---

**Description**

A symbolic routine to define that the systematic component $\eta$ is increasing and concave in a predictor in a formula argument to cgam. This is the unsmoothed version.

**Usage**

```r
incr.conc(x, numknots = 0, knots = 0, space = "Q")
```

**Arguments**

- **x**  
  A numeric predictor which has the same length as the response vector.

- **numknots**  
  The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.

- **knots**  
  The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.

- **space**  
  A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "Q".

**Details**

"incr.conc" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "incr.conc"; the shape attribute is 7("increasing and concave"), and according to the value of the vector itself and its attributes, the cone edges of the cone generated by the constraint matrix, which constrains the relationship between the systematic component $\eta$ and "x" to be increasing and concave, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "incr.conc" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.
Value
The vector x with four attributes, i.e., name: the name of x; shape: 7("increasing and concave"); numknots: the numknots argument in "incr.conc"; knots: the knots argument in "incr.conc".

Author(s)
Mary C. Meyer and Xiyue Liao

References

See Also
incr.conv

Examples
```r
data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x with the shape restriction: "increasing" and "concave"
ans <- cgam(y ~ incr.conc(x))

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "increasing and concave fit", col = 2, lty = 1)
```

Description
A symbolic routine to define that the systematic component $\eta$ is increasing and convex in a predictor in a formula argument to cgam. This is the unsmoothed version.

Usage
```
incr.conv(x, numknots = 0, knots = 0, space = "Q")
```
Arguments

\- **x**: A numeric predictor which has the same length as the response vector.
\- **numknots**: The number of knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
\- **knots**: The knots used to smoothly constrain a predictor. The value should be 0 for a shape-restricted predictor without smoothing. The default value is 0.
\- **space**: A character specifying the method to create knots. It will not be used for a shape-restricted predictor without smoothing. The default value is "Q".

Details

"incr.conv" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "incr.conv"; the shape attribute is 5("increasing and convex"), and according to the value of the vector itself and its attributes, the cone edges of the cone generated by the constraint matrix, which constrains the relationship between the systematic component \( \eta \) and "x" to be increasing and convex, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "incr.conv" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 5("increasing and convex"); numknots: the numknots argument in "incr.conv"; knots: the knots argument in "incr.conv".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

incr.conc, incr

Examples

data(cubic)

# extract x
x <- cubic$x
```r
# extract y
y <- cubic$y

# regress y on x with the shape restriction: "increasing" and "convex"
ans <- cgam(y ~ incr.conv(x))

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "increasing and convex fit", col = 2, lty = 1)
```

---

**plasma**  

*A Data Set for Cgam*

---

**Description**

This data set is used for the routine plotpersp. It contains 314 observations of blood plasma, beta carotene measurements along with several covariates. High levels of blood plasma and beta carotene are believed to be protective against cancer, and it is of interest to determine the relationships with covariates.

**Usage**

data(plasma)

**Format**

- **logplasma** A numeric vector of the logarithm of plasma levels.
- **betacaro** A numeric vector of dietary beta carotene consumed mcg per day.
- **bmi** A numeric vector of BMI values.
- **cholest** A numeric vector of cholesterol consumed mg per day.
- **dietfat** A numeric vector of the logarithm of grams of diet fat consumed per day.
- **fiber** A numeric vector of grams of fiber consumed per day.
- **retinol** A numeric vector of retinol consumed per day.
- **smoke** A numeric vector of smoking status (1=Never, 2=Former, 3=Current Smoker).
- **vituse** A numeric vector of vitamin use (1=Yes, fairly often, 2=Yes, not often, 3=No).

**Source**

[http://axon.cs.byu.edu/data/statlib/numeric/plasma_retinol.arff](http://axon.cs.byu.edu/data/statlib/numeric/plasma_retinol.arff)

**Examples**

data(plasma)
plotpersp

Create a 3D Plot for a Cgam Object

Description

Given an object of the cgam class, which has at least two non-parametrically modelled predictors, this routine will make a 3D plot of the fit with a set of two non-parametrically modelled predictors in the formula being the x and y labs. If there are more than two non-parametrically modelled predictors, any other such predictor will be evaluated at the largest value which is smaller than or equal to its median value.

If there is any categorical covariate and if the user specifies the argument categ to be a character representing a categorical covariate in the formula, then a 3D plot with multiple parallel surfaces, which represent the levels of a categorical covariate in an ascending order, will be created; otherwise, a 3D plot with only one surface will be created. Each level of a categorical covariate will be evaluated at its mode.

This routine is an extension of the generic R graphics routine persp.

Usage

plotpersp(object, x1, x2, surface = "mu", categ = NULL, cols = NULL, random = FALSE, x_grid = 20, y_grid = 20, at = "median", xlim = range(x1), ylim = range(x2), zlim = NULL, xlab = NULL, ylab = NULL, zlab = NULL, main = NULL, sub = NULL, th = -40, phi = 15, r = sqrt(3), d = 1, scale = TRUE, expand = 1, border = NULL, ltheta = -135, lphi = 0, shade = NA, box = TRUE, axes = TRUE, nticks = 5, ticktype = "detailed")

Arguments

object
A non-parametrically modelled predictor in a cgam fit.

x1
A non-parametrically modelled predictor in a cgam fit.

x2
A non-parametrically modelled predictor in a cgam fit.

surface
The type of the surface of a 3D plot. If surface == "mu", then the surface of the estimated mean value of a cgam fit will be plotted; if surface == "eta", then the surface of the estimated systematic component value of a cgam fit will be plotted. The default is surface = "mu".

categ
Optional categorical covariate(s) in a cgam fit. If there is any categorical covariate and if the user specifies the argument categ to be a character representing a categorical covariate in the formula, then a 3D plot with multiple parallel surfaces, which represent the levels of a categorical covariate in an ascending order, will be created; otherwise, a 3D plot with only one surface will be created. Each level of a categorical covariate will be evaluated at its mode. The default is categ = NULL.
plotpersp

**cols**
The color(s) of a 3D plot created by plotpersp. If cols == NULL, "white" will be used when there is only one surface in the plot, and a sequence of colors will be used in a fixed order when there are multiple parallel surfaces in the plot. For example, when there are two surfaces, the lower surface will be in the color "peachpuff", and the higher surface will be in the color "lightblue". The default is cols = NULL.

**random**
A logical scalar. If random == TRUE, color(s) for a 3D plot will be randomly chosen from ten colors, namely, "peachpuff", "lightblue", "limegreen", "grey", "wheat", "yellowgreen", "seagreen1", "palegreen", "azure", "whitesmoke"; otherwise, "white" will be used when there is only one surface in the plot, and a sequence of colors will be used in a fixed order when there are multiple parallel surfaces in the plot.

**x_grid**
This is a positive integer specifying how dense the x grid will be. The default is x_grid = 20.

**y_grid**
This is a positive integer specifying how dense the y grid will be. The default is y_grid = 20.

**at**
When a user specifies a set of two non-parametrically modelled predictors to make a 3D plot, any other non-parametrically modelled predictor in the cgam formula will be evaluated at the largest value which is smaller than or equal to its median value.

**xlim**
The xlim argument inherited from the persp routine.

**ylim**
The ylim argument inherited from the persp routine.

**zlim**
The zlim argument inherited from the persp routine.

**xlab**
The xlab argument inherited from the persp routine.

**ylab**
The ylab argument inherited from the persp routine.

**zlab**
The zlab argument inherited from the persp routine.

**main**
The main argument inherited from the persp routine.

**sub**
The sub argument inherited from the persp routine.

**th**
The theta argument inherited from the persp routine.

**phi**
The phi argument inherited from the persp routine.

**r**
The r argument inherited from the persp routine.

**d**
The d argument inherited from the persp routine.

**scale**
The scale argument inherited from the persp routine.

**expand**
The expand argument inherited from the persp routine.

**border**
The border argument inherited from the persp routine.

**ltheta**
The ltheta argument inherited from the persp routine.

**lphi**
The lphi argument inherited from the persp routine.

**shade**
The shade argument inherited from the persp routine.

**box**
The box argument inherited from the persp routine.

**axes**
The axes argument inherited from the persp routine.

**nticks**
The nticks argument inherited from the persp routine.

**ticktype**
The ticktype argument inherited from the persp routine.
The routine plotpersp returns a 3D plot of an object of the cgam class. The $x$ lab and $y$ lab represent a set of non-parametrically modelled predictors used in a cgam formula. The $z$ lab represents the estimated mean value or the estimated systematic component value.

Author(s)
Mary C. Meyer and Xiyue Liao

References

Examples

```r
# Example 1.
data(FEV)

# extract the variables
y <- FEV$FEV
age <- FEV$age
height <- FEV$height
sex <- FEV$sex
smoke <- FEV$smoke

fit = cgam(y ~ incr(age) + incr(height) + sex + smoke, nsim = 0)
fit1 = cgam(y ~ s.incr(age) + s.incr(height) + sex + smoke, nsim = 0)

plotpersp(fit, age, height, x_grid = 10, y_grid = 10, main = "Cgam Increasing Fit",
sub = "Categorical Variable: Sex", categ = "sex")
plotpersp(fit1, age, height, x_grid = 10, y_grid = 10, main = "Cgam Smooth Increasing Fit",
sub = "Categorical Variable: Smoke", categ = "smoke")

# Example 2.
data(plasma)

# extract the variables
y <- plasma$logplasma
bmi <- plasma$bmi
dietfat <- plasma$dietfat
cholest <- plasma$cholest
fiber <- plasma$fiber
betacaro <- plasma$betacaro
retinol <- plasma$retinol
smoke <- plasma$smoke
vituse <- plasma$vituse

fit <- cgam(y ~ s.decr(bmi) + s.decr(dietfat) + s.decr(cholest) + s.incr(fiber) + s.incr(betacaro) + s.incr(retinol) + smoke + vituse)

plotpersp(fit, bmi, dietfat, x_grid = 10, y_grid = 10, th = 120, ylab = "log(dietfat)")
```
Description

A symbolic routine to define that the systematic component $\eta$ is smooth in a predictor in a formula argument to cgam. This is the smooth version.

Usage

$s(x, \text{numknots} = 0, \text{knots} = 0, \text{space} = "Q")$

Arguments

- $x$: A numeric predictor which has the same length as the response vector.
- numknots: The number of knots used to constrain $x$. It will not be used if the user specifies the knots argument. The default is numknots = 0.
- knots: The knots used to constrain $x$. User defined knots will be used when given. Otherwise, numknots and space will be used to create knots. The default is knots = 0.
- space: A character specifying the method to create knots. It will not be used if the user specifies the knots argument. If space == "E", then equally spaced knots will be created; if space == "Q", then a vector of equal $x$ quantiles will be created based on $x$ with duplicate elements removed. The number of knots is numknots when numknots > 0. Otherwise it is of the order $n^{1/7}$. The default is space = "Q".

Details

"s" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "s"; the shape attribute is 17("smooth"). According to the value of the vector itself and its shape, numknots and knots attributes, the cone edges will be made by C-spline basis functions in Meyer (2008). The cone edges are a set of basis employed in the hinge algorithm.

Note that "s" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 17("smooth"); numknots: the numknots argument in "s"; knots: the knots argument in "s".
**Author(s)**

Mary C. Meyer and Xiyue Liao

**References**


**See Also**

`s.incr`, `s.decr`, `s.conc`, `s.conv`, `s.incr.conc`, `s.incr.conv`, `s.decr.conc`, `s.decr.conv`

**Examples**

```r
# generate y
x <- seq(-1, 2, by = 0.1)
n <- length(x)
y <- - x^2 + rnorm(n, .3)

# regress y on x under the shape-restriction: "smooth"
ans <- cgam(y ~ s(x))
kn <- ans$ knots[[1]]

# make a plot
plot(x, y)
lines(x, ans$ muhat, col = 2)
legend("topleft", bty = "n", "smooth fit", col = 2, lty = 1)
legend(1.6, 1.8, bty = "o", "knots", pch = "x")
points(knots, 1:length(knots)*0+min(y), pch = "x")
```

---

**s.conc**

Specify a Smooth and Concave Shape-Restriction in a CGAM Formula

**Description**

A symbolic routine to define that the systematic component $\eta$ is smooth and concave in a predictor in a formula argument to cgam. This is the smooth version.

**Usage**

`s.conc(x, numknots = 0, knots = 0, space = "Q")`
Arguments

x  A numeric predictor which has the same length as the response vector.

numknots  The number of knots used to constrain x. It will not be used if the user specifies the knots argument. The default is numknots = 0.

knots  The knots used to constrain x. User defined knots will be used when given. Otherwise, numknots and space will be used to create knots. The default is knots = 0.

space  A character specifying the method to create knots. It will not be used if the user specifies the knots argument. If space == "E", then equally spaced knots will be created; if space == "Q", then a vector of equal x quantiles will be created based on x with duplicate elements removed. The number of knots is numknots when numknots > 0. Otherwise it is of the order n^{1/7}. The default is space = "Q".

Details

"s.conc" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "s.conc"; the shape attribute is 12("smooth and concave"). According to the value of the vector itself and its shape, numknots and knots attributes, the cone edges will be made by C-spline basis functions in Meyer (2008). The cone edges are a set of basis employed in the hinge algorithm.

Note that "s.conc" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 12("smooth and concave"); numknots: the numknots argument in "s.conc"; knots: the knots argument in "s.conc".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

conc
Examples

# generate y
x <- seq(-1, 2, by = 0.1)
n <- length(x)
y <- -x^2 + rnorm(n, .3)

# regress y on x under the shape-restriction: "smooth and concave"
ans <- cgam(y ~ s.conv(x))
knots <- ans$knots[[1]]

# make a plot
plot(x, y)
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "smooth and concave fit", col = 2, lty = 1)
legend(1.6, 1.8, bty = "o", "knots", pch = "x")
points(knots, 1:length(knots)*@+min(y), pch = "x")

s.conv

Specify a Smooth and Convex Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the systematic component \( \eta \) is smooth and convex in a predictor in a formula argument to cgam. This is the smooth version.

Usage

s.conv(x, numknots = 0, knots = 0, space = "Q")

Arguments

x
A numeric predictor which has the same length as the response vector.

numknots
The number of knots used to constrain \( x \). It will not be used if the user specifies the knots argument. The default is numknots = 0.

knots
The knots used to constrain \( x \). User defined knots will be used when given. Otherwise, numknots and space will be used to create knots. The default is knots = 0.

space
A character specifying the method to create knots. It will not be used if the user specifies the knots argument. If space == "E", then equally spaced knots will be created; if space == "Q", then a vector of equal \( x \) quantiles will be created based on \( x \) with duplicate elements removed. The number of knots is numknots when numknots > 0. Otherwise it is of the order \( n^{1/7} \). The default is space = "Q".
"s.conv" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.
The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots
dimensions are the same as the numknots and knots arguments in "s.conv"; the shape attribute is
11("smooth and convex"). According to the value of the vector itself and its shape, numknots and
knots attributes, the cone edges will be made by C-spline basis functions in Meyer (2008). The cone
edges are a set of basis employed in the hinge algorithm.

Note that "s.conv" does not make the corresponding cone edges itself. It sets things up to a subrou-
tine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 11("smooth and convex");
umknots: the numknots argument in "s.conv"; knots: the knots argument in "s.conv".

Author(s)

Mary C. Meyer and Xiyue Liao

References

Meyer, M. C. (2013b) A simple new algorithm for quadratic programming with applications in


See Also

cov

Examples

# generate y
x <- seq(-1, 2, by = 0.1)
n <- length(x)
y <- x^2 + rnorm(n, 0.3)

# regress y on x under the shape-restriction: "smooth and convex"
ans <- cgam(y ~ s.conv(x))
knots <- ans$knots[[1]]

# make a plot
plot(x, y)
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "smooth and convex fit", col = 2, lty = 1)
legend(1.6, -1, bty = "n", "knots", pch = "x")
points(knots, 1:length(knots)*0+min(y), pch = "x")
s.decr

Specify a Smooth and Decreasing Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the systematic component $\eta$ is smooth and decreasing in a predictor in a formula argument to cgam. This is the smooth version.

Usage

s.decr(x, numknots = 0, knots = 0, space = "Q")

Arguments

x A numeric predictor which has the same length as the response vector.
numknots The number of knots used to constrain $x$. It will not be used if the user specifies the knots argument. The default is numknots = 0.
knots The knots used to constrain $x$. User defined knots will be used when given. Otherwise, numknots and space will be used to create knots. The default is knots = 0.
space A character specifying the method to create knots. It will not be used if the user specifies the knots argument. If space == "E", then equally spaced knots will be created; if space == "Q", then a vector of equal $x$ quantiles will be created based on $x$ with duplicate elements removed. The number of knots is numknots when numknots > 0. Otherwise it is of the order $n^{1/7}$. The default is space = "Q".

Details

"s.decr" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots. The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "s.decr"; the shape attribute is 10("smooth and decreasing"). According to the value of the vector itself and its shape, numknots and knots attributes, the cone edges will be made by I-spline basis functions in Meyer (2008). The cone edges are a set of basis employed in the hinge algorithm. Note that "s.decr" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam. See references cited in this section for more details.

Value

The vector x with four attributes, i.e, name: the name of x; shape: 10("smooth and decreasing"); numknots: the numknots argument in "s.decr"; knots: the knots argument in "s.decr".
Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also
decr

Examples

data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x under the shape-restriction: "smooth and decreasing"
ans <- cgam(y ~ s.decr(x))
knots <- ans$knots[[1]]

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "smooth and decreasing fit", col = 2, lty = 1)
legend(-.3, 8, bty = "o", "knots", pch = "X")
points(knots, 1:length(knots)*0+min(y), pch = "X")

s.decr.conc  Specify a Smooth, Decreasing and Concave Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the systematic component \( \eta \) is smooth, decreasing and concave in a predictor in a formula argument to cgam. This is the smooth version.

Usage

s.decr.conc(x, numknots = 0, knots = 0, space = "Q")
Arguments

x  A numeric predictor which has the same length as the response vector.
numknots  The number of knots used to constrain \( x \). It will not be used if the user specifies
the knots argument. The default is numknots = 0.
knots  The knots used to constrain \( x \). User defined knots will be used when given.
Otherwise, numknots and space will be used to create knots. The default is
knots = 0.
space  A character specifying the method to create knots. It will not be used if the user
specifies the knots argument. If space == "E", then equally spaced knots will be
created; if space == "Q", then a vector of equal \( x \) quantiles will be created based
on \( x \) with duplicate elements removed. The number of knots is numknots when
numknots > 0. Otherwise it is of the order \( n^{1/7} \). The default is space = "Q".

Details

"s.decr.conc" returns the vector "x" and imposes on it four attributes: name, shape, numknots and
knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots
attributes are the same as the numknots and knots arguments in "s.decr.conc"; the shape attribute
is 16("smooth, decreasing and concave"). According to the value of the vector itself and its shape,
numknots and knots attributes, the cone edges will be made by C-spline basis functions in Meyer
(2008). The cone edges are a set of basis employed in the hinge algorithm.

Note that "s.decr.conc" does not make the corresponding cone edges itself. It sets things up to a
subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 16("smooth, decreasing and
conceave"); numknots: the numknots argument in "s.decr.conc"; knots: the knots argument in
"s.decr.conc".

Author(s)

Mary C. Meyer and Xiyue Liao

References

Meyer, M. C. (2013b) A simple new algorithm for quadratic programming with applications in

tics 2(3), 1013–1033.

See Also

decl.conv, decr
Examples

data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x under the shape-restriction: "smooth, decreasing and concave"
ans <- cgam(y ~ s.decr.conv(x))
knots <- ans$knots[[1]]

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "smooth, decreasing and concave fit", col = 2, lty = 1)
legend(1.7, 4, bty = "o", "knots", pch = "x")
points(knots, 1:length(knots)*0+min(y), pch = "x")

s.decr.conv

Specify a Smooth, Decreasing and Convex Shape-Restriction in a
CGAM Formula

Description

A symbolic routine to define that the systematic component $\eta$ is smooth, decreasing and convex in
a predictor in a formula argument to cgam. This is the smooth version.

Usage

s.decr.conv(x, numknots = 0, knots = 0, space = "Q")

Arguments

x
A numeric predictor which has the same length as the response vector.

numknots
The number of knots used to constrain $x$. It will not be used if the user specifies
the knots argument. The default is numknots = 0.

knots
The knots used to constrain $x$. User defined knots will be used when given.
Otherwise, numknots and space will be used to create knots. The default is
knots = 0.

space
A character specifying the method to create knots. It will not be used if the user
specifies the knots argument. If space == "E", then equally spaced knots will be
created; if space == "Q", then a vector of equal $x$ quantiles will be created based
on $x$ with duplicate elements removed. The number of knots is numknots when
numknots > 0. Otherwise it is of the order $n^{1/7}$. The default is space = "Q".
"s.decr.conv" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "s.decr.conv"; the shape attribute is 15("smooth, decreasing and convex"). According to the value of the vector itself and its shape, numknots and knots attributes, the cone edges will be made by C-spline basis functions in Meyer (2008). The cone edges are a set of basis employed in the hinge algorithm.

Note that "s.decr.conv" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 15("smooth, decreasing and convex"); numknots: the numknots argument in "s.decr.conv"; knots: the knots argument in "s.decr.conv".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

decr.conv

Examples

data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x under the shape-restriction: "smooth, decreasing and convex"
ans <- cgam(y ~ s.decr.conv(x))
knots <- ans$knots[[1]]

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
s.incr

Specify a Smooth and Increasing Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the systematic component $\eta$ is smooth and increasing in a predictor in a formula argument to cgam. This is the smooth version.

Usage

s.incr(x, numknots = 0, knots = 0, space = "Q")

Arguments

x

A numeric predictor which has the same length as the response vector.

numknots

The number of knots used to constrain $x$. It will not be used if the user specifies the knots argument. The default is numknots = 0.

knots

The knots used to constrain $x$. User defined knots will be used when given. Otherwise, numknots and space will be used to create knots. The default is knots = 0.

space

A character specifying the method to create knots. It will not be used if the user specifies the knots argument. If space == "E", then equally spaced knots will be created; if space == "Q", then a vector of equal $x$ quantiles will be created based on $x$ with duplicate elements removed. The number of knots is numknots when numknots > 0. Otherwise it is of the order $n^{1/7}$. The default is space = "Q".

Details

"s.incr" returns the vector "$x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "s.incr"; the shape attribute is 9(“smooth and increasing”). According to the value of the vector itself and its shape, numknots and knots attributes, the cone edges will be made by I-spline basis functions in Meyer (2008). The cone edges are a set of basis employed in the hinge algorithm.

Note that "s.incr" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.
Value

The vector x with four attributes, i.e., name: the name of x; shape: 9("smooth and increasing"); numknots: the numknots argument in "s.incr"; knots: the knots argument in "s.incr".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

incr

Examples

data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x with the shape restriction: "smooth and increasing"
ans <- cgam(y ~ s.incr(x))
knots <- ans$knots[[1]]

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "smooth and increasing fit", col = 2, lty = 1)
legend(1.7, 9.2, bty = "o", "knots", pch = "X")
points(knots, 1:length(knots)*0+min(y), pch = "X")
Usage

s.incr.conc(x, numknots = 0, knots = 0, space = "Q")

Arguments

- **x**: A numeric predictor which has the same length as the response vector.
- **numknots**: The number of knots used to constrain \( x \). It will not be used if the user specifies the **knots** argument. The default is numknots = 0.
- **knots**: The knots used to constrain \( x \). User defined knots will be used when given. Otherwise, **numknots** and **space** will be used to create knots. The default is knots = 0.
- **space**: A character specifying the method to create knots. It will not be used if the user specifies the **knots** argument. If space == "E", then equally spaced knots will be created; if space == "Q", then a vector of equal \( x \) quantiles will be created based on \( x \) with duplicate elements removed. The number of knots is **numknots** when **numknots** > 0. Otherwise it is of the order \( n^{1/7} \). The default is space = "Q".

Details

"s.incr.conc" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "s.incr.conc"; the shape attribute is 14("smooth, increasing and concave"). According to the value of the vector itself and its shape, numknots and knots attributes, the cone edges will be made by C-spline basis functions in Meyer (2008). The cone edges are a set of basis employed in the hinge algorithm.

Note that "s.incr.conc" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 14("smooth, increasing and concave"); numknots: the numknots argument in "s.incr.conc"; knots: the knots argument in "s.incr.conc".

Author(s)

Mary C. Meyer and Xiyue Liao

References


s.incr.conv

Specify an Smooth, Increasing and Convex Shape-Restricion in a CGAM Formula

Description
A symbolic routine to define that the systematic component \( \eta \) is smooth, increasing and convex in a predictor in a formula argument to cgam. This is the smooth version.

Usage
s.incr.conv(x, numknots = 0, knots = 0, space = "Q")

Arguments
- \( x \): A numeric predictor which has the same length as the response vector.
- \( \text{numknots} \): The number of knots used to constrain \( x \). It will not be used if the user specifies the \( \text{knots} \) argument. The default is numknots = 0.
- \( \text{knots} \): The knots used to constrain \( x \). User defined knots will be used when given. Otherwise, \( \text{numknots} \) and \( \text{space} \) will be used to create knots. The default is knots = 0.

Examples
```r
data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y

# regress y on x with the shape restriction: "smooth, increasing and concave"
ans <- cgam(y ~ s.incr.conv(x))
knots <- ans$ knots[[1]]

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$ muhat, col = 2)
legend("topleft", bty = "n", "smooth, increasing and concave fit", col = 2, lty = 1)
legend(-.3, 4, bty = "o", "knots", pch = "X")
points(knots, 1:length(knots)*0+min(y), pch = "X")
```
space

A character specifying the method to create knots. It will not be used if the user specifies the knots argument. If space == "E", then equally spaced knots will be created; if space == "Q", then a vector of equal x quantiles will be created based on x with duplicate elements removed. The number of knots is numknots when numknots > 0. Otherwise it is of the order n^{1/7}. The default is space = "Q".

Details

"s.incr.conv" returns the vector "x" and imposes on it four attributes: name, shape, numknots and knots.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the numknots and knots attributes are the same as the numknots and knots arguments in "s.incr.conv"; the shape attribute is 13("smooth, increasing and convex"). According to the value of the vector itself and its shape, numknots and knots attributes, the cone edges will be made by C-spline basis functions in Meyer (2008). The cone edges are a set of basis employed in the hinge algorithm.

Note that "s.incr.conv" does not make the corresponding cone edges itself. It sets things up to a subroutine called makedelta in cgam.

See references cited in this section for more details.

Value

The vector x with four attributes, i.e., name: the name of x; shape: 13("smooth, increasing and convex"); numknots: the numknots argument in "s.incr.conv"; knots: the knots argument in "s.incr.conv".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

incr.conv

Examples

data(cubic)

# extract x
x <- cubic$x

# extract y
y <- cubic$y
```
# regress y on x with the shape restriction: "smooth, increasing and convex"
ans <- cgam(y ~ s.incr.conv(x))
knots <- ans$knots[[1]]

# make a plot
par(mar = c(4, 4, 1, 1))
plot(x, y, cex = .7, xlab = "x", ylab = "y")
lines(x, ans$muhat, col = 2)
legend("topleft", bty = "n", "smooth, increasing and convex fit", col = 2, lty = 1)
legend(1.7, 9.2, bty = "o", "knots", pch = "x")
points(knots, 1:length(knots)*@+min(y), pch = "x")
```

```

tree Specify a Tree-Ordering in a CGAM Formula

Description
A symbolic routine to define that the systematic component \( \eta \) has a tree-ordering in a predictor in a formula argument to cgam.

Usage
tree(x)

Arguments

x A numeric vector which has the same length as the response vector. Note that the placebo level of x must be 0.

Details
"tree" returns the vector "x" and imposes on it two attributes: name and shape.
The name attribute is used in the subroutines: predict.cgam and plotpersp; the shape attribute is "tree", and according to the value of the vector itself and its shape attribute, the cone edges of the cone generated by the constraint matrix, which constrains that \( \eta \) has a tree-ordering in "x" will be made. The cone edges are a set of basis employed in the hinge algorithm.
Note that "tree" does not make the corresponding cone edges itself. It sets things up to a sub-routine called tree.fun in cgam which will make the cone edges. A tree-ordering is a partial ordering: For a categorical variable x, if there are treatment levels \( x_1, \ldots, x_k \), where \( x_1 \) is a placebo, we compare \( x_i, i = 2, \ldots, k \) with \( x_1 \), and not have any other comparable pairs.
See references cited in this section for more details.

Value
The vector x with two attributes, i.e., name: the name of x; shape: "tree".
```
Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

umbrella

Examples

```r
# generate y
set.seed(123)
n <- 12
x <- rep(0:2, each = 4)
y <- x + rnorm(n, .1)

# regress y on x under the tree-ordering restriction
ans <- cgam(y ~ tree(x))

# make a plot
par(mar = c(4, 4, 1, 1))
plot(y, cex = .7, ylab = "y")
lines(ans$muhat, col = 2)
legend("topleft", bty = "n", "tree-ordering fit", col = 2, lty = 1)
```

---

**umbrella**

*Specify an Umbrella-Ordering in a CGAM Formula*

Description

A symbolic routine to define that the systematic component $\eta$ has an umbrella-ordering in a predictor in a formula argument to cgam.

Usage

`umbrella(x)`

Arguments

- **x**: A numeric vector which has the same length as the response vector.
Details

"umbrella" returns the vector "x" and imposes on it two attributes: name and shape.

The name attribute is used in the subroutines: predict.cgam and plotpersp; the shape attribute is "umbrella", and to the value of the vector itself and its shape attribute, the cone edges of the cone generated by the constraint matrix, which constrains that \( \eta \) has an umbrella-ordering in "x" will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "umbrella" does not make the corresponding cone edges itself. It sets things up to a subroutine called umbrella.fun in cgam which will make the cone edges. An umbrella-ordering is a partial ordering: Suppose we have a \( x_0 \) that is known to be a "mode" so that for \( x, y \geq x_0 \), we have a binary relation between \( x \) and \( y \) if \( x \leq y \) and for \( x, y \leq x_0 \) we have the same binary relation if \( x \leq y \), but if \( x < x_0 \) and \( y > x_0 \), there is no such binary relation.

See references cited in this section for more details.

Value

The vector x with two attributes, i.e., name: the name of x; shape: "umbrella".

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

tree

Examples

```r
# generate y
set.seed(123)
n <- 20
x <- seq(-2, 2, length = n)
y <- x^2 + rnorm(n)

# regress y on x under the umbrella-ordering restriction
ans <- cgam(y ~ umbrella(x))

# make a plot
par(mar = c(4, 4, 1, 1))
plot(y, cex = .7, ylab = "y")
lines(ans$muhat, col = 2)
legend("topleft", bty = "n", "umbrella-ordering fit", col = 2, lty = 1)
```
Index

*Topic **3D plot routine**
plotpersp, 22

*Topic **cgam routine**
cgam, 2

*Topic **data set**
cubic, 11

*Topic **datasets**
plasma, 21

*Topic **shape routine**
  conc, 8
  conv, 10
  decr, 12
  decr.conc, 13
  decr.conv, 15
  incr, 16
  incr.conc, 18
  incr.conv, 19
  s, 25
  s.conc, 26
  s.conv, 28
  s.decr, 30
  s.decr.conc, 31
  s.decr.conv, 33
  s.incr, 35
  s.incr.conc, 36
  s.incr.conv, 38
  tree, 40
  umbrella, 41

cgam, 2
conc, 3, 8, 11, 27
conv, 3, 9, 10, 29
cubic, 11
deincr, 2, 12, 14, 16, 31, 32
deincr.conc, 3, 13, 13, 16
deincr.conv, 3, 13, 14, 15, 32, 34

incr, 2, 16, 20, 36
incr.conc, 3, 17, 18, 20
incr.conv, 3, 17, 19, 19, 38, 39
plasma, 21
plotpersp, 22
s, 3, 25
s.conc, 3, 26, 26
s.conv, 3, 26, 28
s.decr, 3, 26, 30
s.decr.conc, 3, 26, 31
s.decr.conv, 3, 26, 33
s.incr, 2, 26, 35
s.incr.conc, 3, 26, 36
s.incr.conv, 3, 26, 38

tree, 3, 40, 42
umbrella, 3, 41, 41

43