Package ‘cgam’

March 5, 2014

Type Package

Title Constrained Generalized Additive Model

Version 1.0

Date 2014-03-05

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) is used to select the best combination of variables and shapes. This package depends on the R package coneproj, in which the hinge algorithm for one cone projection is effected.

License GPL (>= 2)

Depends coneproj(>= 1.2), R(>= 2.15.2)

NeedsCompilation no

Suggests stats, MASS, graphics, grDevices, utils

Repository CRAN

Date/Publication 2014-03-05 18:14:53

R topics documented:

cgam ................................................................. 2
conc ............................................................... 7
conv ............................................................... 8

cubic .............................................................. 10
decr .............................................................. 10
decr.conc ......................................................... 11
Description

When the user provides a set of predictors with shape or order restriction, and a set of categorical covariates without any shape or order restriction, the cgam routine will return the maximum likelihood estimator for the constrained generalized additive model, using an iteratively re-weighted cone projection algorithm. The coneB routine in the R package coneproj is used in cgam to effect the hinge algorithm for one cone projection.

Usage

cgam(formula, nsim = 0, 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 error distribution of the model can be one of the three exponential families: gaussian, binomial and poisson. A predictor can either be a non-parametrically modelled variable with a shape or order restriction, or a parametrically modelled unconstrained categorical covariate. The user can include both types of predictors in a cgam model. In terms of the shape or order restriction on a non-parametrically modelled predictor, the user is supposed to indicate the restriction by the following way:

Assume that \( y \) is the response and \( x \) is a predictor:

- incr(x): \( y \) is increasing in \( x \).
- decr(x): \( y \) is decreasing in \( x \).
- conc(x): \( y \) is concave in \( x \).
- conv(x): \( y \) is convex in \( x \).
- incr.conc(x): \( y \) is increasing and concave in \( x \).
- decr.conc(x): \( y \) is decreasing and concave in \( x \).
- incr.conv(x): \( y \) is increasing and convex in \( x \).
- decr.conv(x): \( y \) is decreasing and convex in \( x \).
- tree(x): \( y \) has a tree-ordering in \( x \).
- umbrella(x): \( y \) has an umbrella-ordering in \( x \).
The number of simulations used to get the edf0 parameter. Note that when there is no shape-restricted or order-restricted predictor, nsim should be 0. The default is nsim = 0.

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 glm routine in the stats package. There are three families used in cgam: gaussian, binomial and poisson.

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

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

We consider generalized partial linear models with independent observations from an exponential family of the form

\[ p(y_i; \theta, \tau) = \exp\left\{ y_i \theta_i - b(\theta_i) \right\} \tau - c(y_i, \tau), \quad i = 1, \ldots, n \]

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 \( g(\mu_i) = \eta_i, i = 1, \ldots, n \), 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 \( x_l \), \( \beta \) 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, and the solution is obtained through an iteratively re-weighted cone projection, with no back-fitting of individual components.

Suppose that \( \eta \) is an \( n \) by 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 \( n \) by \( p \) design matrix for the parametrically modelled predictors.

The constraints for the component \( \phi_l \) are in \( C_l \), where \( C_l = \{ \phi : A_l\phi \geq 0 \text{ and } B_l\phi = 0 \} \), for matrices \( A_l \) and \( B_l \).

The set \( C_l \) is a convex cone and the set \( C = C_1 + \ldots + C_p + Z \) is also a convex cone with a finite set of edges, where the edges are the generator 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 for the R package coneproj for more details.

The fitted value of the response on the null space of the constraint set.

The fitted systematic component \( \eta \).
muhat The fitted mean value, obtained by transforming the systematic component $\eta$ by the inverse of the link function.

vcoefs The estimated coefficients for the basis spanning the null space of the constraint set.

xcoefs The estimated coefficients for the edges corresponding to the shape-restricted predictors.

zcoefs The estimated coefficients for zmat, i.e., the estimation for the vector $\beta$.

ucoefs The estimated coefficients for the edges corresponding to the predictors with an umbrella-ordering constraint.

tcoefs The estimated coefficients for the edges corresponding to the predictors with a tree-ordering constraint.

cols The estimated coefficients for the basis spanning the null space of the constraint set and edges corresponding to the constrained predictors.

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.

etacomps The fitted mean value for the shape-restricted or order-restricted predictors. It is a matrix of which each row is the fitted mean value for a shape-restricted or order-restricted predictor. If there are more than one such predictor, the order of the rows is the same as the order that the user defines shape-restricted or order-restricted predictors in the formula argument of cgam.

xmat A matrix whose columns represent the shape-restricted predictors.

zmat A matrix whose columns represent the unconstrained 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 represnt the predictors with a tree-ordering constraint.

umb A matrix whose columns represnt 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 edges corresponding to the constrained predictors.

shapes A vector including the shape constraints in a cgam fit.

wt The weights in the final iteration of the iteratively re-weighted cone projections.

wt.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 by the user in cgam.

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.

df

The degree of freedom for the null space of a cgam fit.

resid_df.obs

The observed degree of freedom for the residuals of a cgam fit.

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.

tms

The terms objects extracted by the generic function `terms` from a cgam fit. See the help page of the `terms` function for more details.

capm

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

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 storing the indices keeping track of the beginning position of the set of edges for each shape-restricted predictor in xmat.

xid2

A vector storing the indices keeping track of the end position of the set of edges for each shape-restricted predictor in xmat.

tid1

A vector storing the indices keeping track of the beginning position of the set of edges for each tree-ordering factor in tr.

tid2

A vector storing the indices keeping track of the end position of the set of edges for each tree-ordering factor in tr.

uid1

A vector storing the indices keeping track of the beginning position of the set of edges for each umbrella-ordering factor in umb.

uid2

A vector storing the indices keeping track of the end position of the set of edges for each umbrella-ordering factor in umb.

call

The matched call.

Author(s)

Mary C. Meyer and Xiyue Liao
References


Examples

```r
# Example 1.
data(cubic)
data 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, fitted(fit.lm), col = 1, lty = 2)
lines(x, fitted(fit.cgam), 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(),
            nsim = 1e+3, data = kyphosis)
```
conc

## Example

```r
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)
n <- 10
hard <- Rubber$hard
tens <- Rubber$tens

# make a 3D plot based on fit.cgam
x1grid <- seq(min(hard), max(hard), length = n)
grid <- seq(min(tens), max(tens), length = n)
twoddata <- data.frame(hard = rep(x1grid, each = length(xgrid)),
tens = rep(xgrid, length(x1grid)))
thint <- predict(fit.cgam, twoddata)fit
A <- matrix(thint, length(x1grid), length(xgrid), byrow = TRUE)
persp(A, xlab = "hard", ylab = "tens", zlab = "loss", theta = 120)
title("3D Plot of a CGAM Fit")
```

---

### conc

**Specify a Concave Shape-Restriction in a CGAM Formula**

#### Description

A symbolic routine to define that the response is concave in a predictor in a formula argument to cgam

#### Usage

```r
conc(x)
```

#### Arguments

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

#### Details

"conc" returns the vector "x" and imposes on it a shape attribute: "concave". 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 response 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 a concave shape tag.

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

conv

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: "concave"
ans <- cgam(y ~ conv(x))

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

---

conv

Specify a Convex Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the response is convex in a predictor in a formula argument to cgam.

Usage

conv(x)

Arguments

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

"conv" returns the vector "x" and imposes on it a shape attribute: "convex". 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 response and "x" to be convex, will be made. The cone edges are a set of basis employed in the hinge algorithm.

Note that "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 a convex shape tag.

**Author(s)**

Mary C. Meyer and Xiyue Liao

**References**


**See Also**

conc

**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: "convex"
ans <- cgam(y ~ conv(x))

# 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 response is decreasing in a predictor in a formula argument to cgam.

Usage
decr(x)

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

Details
"decr" returns the vector "x" and imposes on it a shape attribute: "decreasing". 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 response 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.
decr.conc

Value

The vector x with a decreasing shape tag.

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)
Arguments

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

Details

"decr.conc" returns the vector "x" and imposes on it a shape attribute: "decreasing and concave". 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 response 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 a decreasing and concave shape tag.

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

decr.conv, decr

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

Description

A symbolic routine to define that the response is decreasing and convex in a predictor in a formula argument to cgam.

Usage

decr.conv(x)

Arguments

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

Details

"decr.conv" returns the vector "x" and imposes on it a shape attribute: "decreasing and convex". 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 response 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 a decreasing and convex shape tag.

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

decr.conc, decr
Examples

```r
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 response is increasing in a predictor in a formula argument to cgam.

Usage

```r
incr(x)
```

Arguments

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

Details

"incr" returns the vector "x" and imposes on it a shape attribute: "increasing". 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 response 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 an increasing shape tag.
Author(s)
Mary C. Meyer and Xiyue Liao

References

See Also
incr.conc, incr.conv

Examples
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 response is increasing and concave in a predictor in a formula argument to cgam.

Usage
incr.conc(x)

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

"incr.conc" returns the vector "x" and imposes on it a shape attribute: "increasing and concave". 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 response 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 an increasing and concave shape tag.

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: "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)
 Specify an Increasing and Convex Shape-Restriction in a CGAM Formula

Description

A symbolic routine to define that the response is increasing and convex in a predictor in a formula argument to cgam.

Usage

incr.conv(x)

Arguments

x

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

Details

"incr.conv" returns the vector "x" and imposes on it a shape attribute: "increasing and convex". 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 response 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 an increasing and convex shape tag.

Author(s)

Mary C. Meyer and Xiyue Liao

References


See Also

incr.conc, incr
Examples

data(cubic)

# extract x
x <- cubic$x

# 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)

---

**tree**

Specify a Tree-Ordering in a CGAM Formula

Description

A symbolic routine to define that the response 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 a shape attribute: "tree". 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 the response 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 a tree-ordering tag.

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 response 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 a shape attribute: "umbrella". 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 the response 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 an umbrella-ordering tag.

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 cgam routine
cgam, 2
*Topic data set
cubic, 10
*Topic shape routine
conc, 7
conv, 8
decr, 10
decr.conc, 11
decr.conv, 13
incr, 14
incr.conc, 15
incr.conv, 17
tree, 18
umbrella, 19

cgam, 2
conc, 7, 9
conv, 8, 8
cubic, 10
decr, 10, 12, 13
decr.conc, 11, 11, 13
decr.conv, 11, 12, 13
incr, 14, 17
incr.conc, 15, 15, 17
incr.conv, 15, 16, 17
tree, 18, 20
umbrella, 19, 19