

Package ‘rgam’

February 20, 2015

Type Package

Title Robust Generalized Additive Model

Version 0.6.3

Date 2014-01-31

Author Matias Salibian-Barrera, Davor Cubranic, and Azadeh Alimadad, with 'loess' code from Cleveland, Grosse, and Shyu

Maintainer Davor Cubranic <cubranic@stat.ubc.ca>

Description Robust Generalized Additive Model

URL <http://www.stat.ubc.ca/~matias/rgam/>

License GPL (>= 3)

LazyLoad yes

LazyData yes

Depends R (>= 3.0.2), stats

Imports Rcpp (>= 0.8.0)

LinkingTo Rcpp (>= 0.8.0), RcppArmadillo (>= 0.2.1)

SystemRequirements GNU make

Collate 'rgam.R'

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-01-31 21:47:21

R topics documented:

rgam-package	2
ili.visits	2
predict.rgam	3
rgam	4

Index	7
--------------	----------

rgam-package	Package 'rgam'
--------------	----------------

Description

The rgam package is an implementation of algorithms for outlier-robust fit for Generalized Additive Models described in *Azadeh and Salibian-Barrera (2011)*. It has one entry function, `rgam`. For more details refer to the relevant help files.

Author(s)

Matias Salibian-Barrera and Davor Cubranic.

Maintainer: Davor Cubranic <cubranic@stat.ubc.ca>

References

Azadeh, A. and Salibian-Barrera, M. (2011). *An outlier-robust fit for Generalized Additive Models with applications to disease outbreak detection*. To appear in the Journal of the American Statistical Association.

Examples

```
## see examples for rgam and ili.visits.
```

ili.visits	<i>Influenza-like-illness outpatient visits.</i>
------------	--

Description

These data contain the number of reported influenza-like-illness outpatient visits in the US for the 2006, 2007 and 2008 seasons. A season consists of weeks 40 to week 20 of the following year, so that, for example, the 2008 season includes up to week 20 of 2009, where the H1N1 flu epidemic had already caused a noticeable increase in ILI visits.

Usage

```
ili.visits
```

Format

A data frame with 99 observations on the following 3 variables.

`visits` the number of reported ILI visits for that week (numeric)

`season` the season year (numeric)

`week` the week number for that season (numeric)

Source

US Centre for Disease Control, <http://www.cdc.gov/flu/weekly/fluactivity.htm>

Examples

```
library(rgam)
plot(visits ~ week, data=ili.visits, xlab='Week', ylab='ILI visits',
     pch=19, col='grey75')
with(ili.visits, {
  a <- rgam(x=week, y=visits, family='poisson', cv.method='rcv',
           epsilon=1e-7, alpha=17/80, max.it=500)
  pr.rgam <- predict(a, type='response')
  lines(week[order(week)], pr.rgam[order(week)], lwd=3, col='red')
})
```

predict.rgam

Predict method for RGAM fits

Description

Obtains predictions from a robustly fitted generalized additive model object

Usage

```
## S3 method for class 'rgam'
predict(object, type=c("link", "response", "terms"), ...)
```

Arguments

object	a fitted rgam object
type	a character string specifying the type of predictions. Can be one of 'link' (the default), 'response', or 'terms'.
...	additional arguments passed from other methods

Details

Serves as the extractor function on objects of class rgam.

Value

the component of the 'object' based on the value of type: if the 'code' is 'response', returns the \$fitted.values; if the type is 'link', returns the \$additive.predictors; and if the type is 'terms', returns the \$smooth component.

Author(s)

Matias Salibian-Barrera <matias@stat.ubc.ca> and Davor Cubranic <cubranic@stat.ubc.ca>

Description

rgam is used to obtain an outlier-robust fit for generalized additive models. It uses the backfitting algorithm with weights derived from robust quasi-likelihood equations. Currently, only local regression smoothers are supported. Bandwidth selection using robust and non-robust cross-validation criteria is currently only implemented for models with a single covariate.

Usage

```
rgam(x, y, family=c("poisson", "binomial"), ni, epsilon=1e-08,
     max.it=50, k=1.5, trace=FALSE, cv.method=c("rcv", "cv", "dcv",
     "rdcv"), alpha=seq(0.1, 0.9, by = 0.1), s.i)
```

Arguments

x	a vector or matrix of covariates
y	a vector of responses
family	a character string indicating the assumed distribution of the response (conditional on the covariates). Only 'poisson' and 'binomial' are implemented. The link function is currently chosen to be the canonical link for the selected family (log for 'poisson' and logit for 'binomial')
ni	a vector of the same length as y containing the number of tries of the binomial distribution of each entry of y. Only relevant if the argument family equals 'binomial'
epsilon	tolerance for the convergence of the robust local scoring algorithm
max.it	maximum number of robust local scoring iterations
k	tuning constant for the robust quasi-likelihood score equations. Large values of k make the estimators closer to the classical fit (and hence less robust), while smaller values of k produce a more robust fit. Values between 1.5 and 3 generally result in a fit with good robustness properties
trace	logical flag to turn on debugging output
cv.method	character string indicating which cross-validation criterion is to be mimized to select the bandwidth from the list given in the argument alphas. Accepted values are 'rcv' (for a weighted squared loss where the effect of outliers is reduced); 'cv' (for the "classical" squared loss); 'dcv' (for the classical deviance loss); 'rdcv' (for a robustly weighted deviance loss). See the references for more details
alpha	a scalar (for models with a single covariate it can be a vector of numbers) between 0 and 1. If length(alphas)==1, its value is used as bandwidth for the local regression smoother, as described in loess. If alphas is a vector, then the value that minimizes the cross-validation criterion specified in the argument 'cv' is used.

`s.i` optional matrix of initial values for the additive predictors (including the intercept). If missing the predictors are initialized at zero and the intercept is taken to be the transformed sample mean of the responses.

Details

The `gam` model is fit using the robust local scoring algorithm, which iteratively fits weighted additive models by backfitting. The weights are derived from robust quasi-likelihood estimating equations and thus effectively reduce the potentially damaging effect of outliers.

Currently, this function only implements local regression smoothers (as calculated by `loess`). The method can be applied to other smoothers as well.

Value

returns an object of class `rgam`. It contains the following components:

<code>additive.predictors</code>	the additive fit, the sum of the columns of the <code>\$smooth</code> component
<code>fitted.values</code>	the fitted mean values, obtained by transforming the component <code>'additive.predictors'</code> using the inverse link function
<code>smooth</code>	the matrix of smooth terms, columns correspond to the smooth predictors in the model
<code>iterations</code>	number of robust local scoring iterations used
<code>convergence.criterion</code>	last relative change of the additive predictors
<code>converged</code>	a logical value indicating whether the algorithm stopped due to the relative change of consecutive additive predictors being less than the tolerance specified in the <code>epsilon</code> argument (TRUE) or because the maximum number of iterations (in the argument <code>max.it</code>) was reached (FALSE)
<code>alpha</code>	the candidate bandwidth values that were considered
<code>cv.method</code>	a character string indicating the cross-validation method used to choose the bandwidth of the smoother
<code>cv.results</code>	a vector of the cross-validation criteria values obtained with each entry of the argument <code>alpha</code>
<code>opt.alpha</code>	the value in the argument <code>alpha</code> that produced the smallest cross-validation criterion. This is the bandwidth used for the reported fit.

Author(s)

Matias Salibian-Barrera <matias@stat.ubc.ca> and Davor Cubranic <cubranic@stat.ubc.ca>

References

Azadeh, A. and Salibian-Barrera, M. (2011). *An outlier-robust fit for Generalized Additive Models with applications to disease outbreak detection*. To appear in the Journal of the American Statistical Association.

Examples

```
x <- ili.visits$week
y <- ili.visits$visits
set.seed(123)
x <- x + rnorm(x, mean=0, sd=.01)
#
# the following command needs to run over 890 fits
# and takes about 22 mins on an Intel Xeon CPU (3.2GHz)
#
# a <- rgam(x=x, y=y, family='poisson', cv.method='rcv',
# epsilon=1e-5, alpha=12:20/80, max.it=500)
#
# the optimal is found at alpha = 17/80
#
a <- rgam(x=x, y=y, family='poisson', cv.method='rcv',
epsilon=1e-7, alpha=17/80, max.it=500)

pr.rgam.a <- predict(a, type='response')
plot(x, y, xlab='Week', ylab='ILI visits', pch=19, col='grey75')
lines(x[order(x)], pr.rgam.a[order(x)], lwd=3, col='red')
```

Index

- *Topic **datasets**
 - [ili.visits, 2](#)
- *Topic **models,**
 - [rgam, 4](#)
- *Topic **package**
 - [rgam-package, 2](#)
- *Topic **regression,**
 - [rgam, 4](#)
- *Topic **robust,**
 - [rgam, 4](#)
- *Topic **robust**
 - [rgam-package, 2](#)
- *Topic **smooth**
 - [rgam, 4](#)

[ili.visits, 2](#)

[predict.rgam, 3](#)

[rgam, 2, 4](#)

[rgam-package, 2](#)