Package 'RBF'

Type Package

Title Robust Backfitting

Version 2.0.1

Date 2020-07-29

Description A robust backfitting algorithm for additive models based on (robust) local polynomial kernel smoothers. It includes both bounded and re-descending (kernel) M-estimators, and it computes predictions for points outside the training set if desired. See Boente, Martinez and Salibian-Barrera (2017) <doi:10.1080/10485252.2017.1369077> for details.

License GPL (>= 3.0)

RoxygenNote 6.1.1

Encoding UTF-8

Imports stats, graphics

NeedsCompilation yes

Author Matias Salibian-Barrera [aut, cre],
Alejandra Martinez [aut]

Maintainer Matias Salibian-Barrera <matias@stat.ubc.ca>

Repository CRAN

Date/Publication 2020-07-30 04:40:02 UTC

R topics documented:

RBF-package   *A robust backfitting algorithm for additive models.*

Description

A robust backfitting algorithm for additive models.

Details

Package:	RBF
Type:	Package
Version:	1.0
Date:	2015-01-19
License:	GPL 3.0

Author(s)

Matias Salibian-Barrera, Alejandra Martinez

Maintainer: Matias Salibian-Barrera <matias@stat.ubc.ca>

References

Boente G, Martinez A, Salibian-Barrera M. Robust estimators for additive models using backfitting. Journal of Nonparametric Statistics, 2017; 29:744-767. https://doi.org/10.1080/10485252.2017.1369077

backf.cl   *Classic Backfitting*

Description

This function computes the standard backfitting algorithm for additive models.

Usage

```
backf.cl(formula, data, subset, point = NULL, windows, epsilon = 1e-06,
  degree = 0, prob = NULL, max.it = 100)
```

Arguments

formula	an object of class formula (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by [as.data.frame](#) to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which the function was called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
point	matrix of points where predictions will be computed and returned.
windows	vector of bandwidths for the local polynomial smoother, one per explanatory variable.
epsilon	convergence criterion. Maximum allowed relative difference between consecutive estimates
degree	degree of the local polynomial smoother. Defaults to 0 (local constant).
prob	vector of probabilities of observing each response (length n). Defaults to NULL and in that case it is ignored.
max.it	Maximum number of iterations for the algorithm.

Details

This function computes the standard backfitting algorithm for additive models, using a squared loss function and local polynomial smoothers.

Value

A list with the following components:

alpha	Estimate for the intercept.
g.matrix	Matrix of estimated additive components (n by p).
prediction	Matrix of estimated additive components for the points listed in the argument point.

Author(s)

Matias Salibian-Barrera, <matias@stat.ubc.ca>, Alejandra Martinez

References

Hasie, TJ and Tibshirani, RJ. Generalized Additive Models, 1990. Chapman and Hall, London.

Examples

```
data(airquality)
tmp <- backf.cl(Ozone ~ Solar.R + Wind + Temp, data=airquality,
subset=complete.cases(airquality), windows=c(130, 9, 10), degree=1)
```

| backf.rob | *Robust Backfitting* |

Description

This function computes a robust backfitting algorithm for additive models

Usage

```
backf.rob(formula, data, subset, windows, point = NULL,
  epsilon = 1e-06, degree = 0, sigma.hat = NULL, prob = NULL,
  max.it = 50, k.h = 1.345, k.t = 4.685, type = "Huber")
```

Arguments

formula	an object of class `formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which the function was called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
windows	vector of bandwidths for the local polynomial smoother, one per explanatory variable.
point	matrix of points where predictions will be computed and returned.
epsilon	convergence criterion. Maximum allowed relative difference between consecutive estimates
degree	degree of the local polynomial smoother. Defaults to `0` (local constant).
sigma.hat	estimate of the residual standard error. If NULL (default) we use the mad of the residuals obtained with local medians.
prob	vector of probabilities of observing each response (length n). Defaults to NULL and in that case it is ignored.
max.it	Maximum number of iterations for the algorithm.
k.h	tuning constant for a Huber-type loss function.
k.t	tuning constant for a Tukey-type loss function.
type	one of either `'Tukey'` or `'Huber'`.

Details

This function computes a robust backfitting algorithm for additive models using robust local polynomial smoothers.

Value

A list with the following components:

alpha	Estimate for the intercept.
g.matrix	Matrix of estimated additive components (n by p).
prediction	Matrix of estimated additive components for the points listed in the argument point.
sigma.hat	Estimate of the residual standard error.

Author(s)

Matias Salibian-Barrera, <matias@stat.ubc.ca>, Alejandra Martinez

References

Boente G, Martinez A, Salibian-Barrera M. Robust estimators for additive models using backfitting. Journal of Nonparametric Statistics, 2017; 29:744-767. https://doi.org/10.1080/10485252.2017.1369077

Examples

```
data(airquality)
tmp <- backf.rob(Ozone ~ Solar.R + Wind + Temp, data=airquality,
subset=complete.cases(airquality), windows=c(136.7, 8.9, 4.8), degree=1)
```

---

deviance.backf	*Deviance for objects of class* backf

Description

This function returns the deviance of the fitted additive model using one of the three classical or robust marginal integration estimators, as computed with backf.cl or backf.rob.

Usage

```
## S3 method for class 'backf'
deviance(object, ...)
```

Arguments

object	an object of class backf, a result of a call to backf.cl or backf.rob.
...	additional other arguments. Currently ignored.

Value

A real number.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

---

fitted.values.backf   *Fitted values for objects of class* backf

---

Description

This function returns the fitted values given the covariates of the original sample under an additive model using a classical or robust marginal integration procedure estimator computed with backf.cl or backf.rob.

Usage

```
fitted.values.backf(object, ...)
```

Arguments

object	an object of class backf, a result of a call to backf.cl or backf.rob.
...	additional other arguments. Currently ignored.

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

---

formula.backf   *Additive model formula*

Description

Description of the additive model formula extracted from an object of class backf.

Usage

```
## S3 method for class 'backf'
formula(x, ...)
```

Arguments

x	an object of class backf, a result of a call to backf.cl or backf.rob.
...	additional other arguments. Currently ignored.

Value

A model formula.

Author(s)

Alejandra Mercedes Martinez `<ale_m_martinez@hotmail.com>`

---

k.epan   *Epanechnikov kernel*

Description

This function evaluates an Epanechnikov kernel

Usage

```
k.epan(x)
```

Arguments

x   a vector of real numbers

Details

This function evaluates an Epanechnikov kernel

Value

A vector of the same length as x where each entry is `0.75 * (1 -x^2)` if x < 1 and 0 otherwise.

Author(s)

Matias Salibian-Barrera, `<matias@stat.ubc.ca>`, Alejandra Martinez

Examples

```
x <- seq(-2, 2, length=10)
k.epan(x)
```

---

plot.backf   *Diagnostic plots for objects of class* backf

Description

Plot method for objects of class backf.

Usage

```
## S3 method for class 'backf'
plot(x, ask = FALSE, which = 1:np, ...)
```

Arguments

x	an object of class backf, a result of a call to [backf.cl](#) or [backf.rob](#).
ask	logical value. If TRUE, the graphical device will prompt for confirmation before going to the next page/screen of output.
which	vector of indices of explanatory variables for which partial residuals plots will be generaetd. Defaults to all available explanatory variables.
...	additional other arguments. Currently ignored.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

Examples

```
tmp <- backf.rob(Ozone ~ Solar.R + Wind + Temp, data=airquality,
subset=complete.cases(airquality), windows=c(136.7, 8.9, 4.8), degree=1)
plot(tmp, which=1:2)
```

---

predict.backf   *Fitted values for objects of class* backf.

Description

This function returns the fitted values given the covariates of the original sample under an additive model using the classical or robust backfitting approach computed with [backf.cl](#) or [backf.rob](#).

Usage

```
## S3 method for class 'backf'
predict(object, ...)
```

Arguments

object	an object of class backf, a result of a call to [backf.cl](#) or [backf.rob](#).
...	additional other arguments. Currently ignored.

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

---

print.backf   *Print a Marginal Integration procedure*

Description

The default print method for a `backf` object.

Usage

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

Arguments

x	an object of class backf, a result of a call to [backf.cl](#) or [backf.rob](#).
...	additional other arguments. Currently ignored.

Value

A real number.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

---

psi.huber   *Derivative of Huber's loss function.*

Description

This function evaluates the first derivative of Huber's loss function.

Usage

```
psi.huber(r, k = 1.345)
```

Arguments

r   a vector of real numbers

k   a positive tuning constant.

Details

This function evaluates the first derivative of Huber's loss function.

Value

A vector of the same length as x.

Author(s)

Matias Salibian-Barrera, <matias@stat.ubc.ca>, Alejandra Martinez

Examples

```
x <- seq(-2, 2, length=10)
psi.huber(r=x, k = 1.5)
```

---

psi.tukey   *Derivative of Tukey's bi-square loss function.*

Description

This function evaluates the first derivative of Tukey's bi-square loss function.

Usage

```
psi.tukey(r, k = 4.685)
```

Arguments

r	a vector of real numbers
k	a positive tuning constant.

Details

This function evaluates the first derivative of Tukey's bi-square loss function.

Value

A vector of the same length as x.

Author(s)

Matias Salibian-Barrera, <matias@stat.ubc.ca>, Alejandra Martinez

Examples

```
x <- seq(-2, 2, length=10)
psi.tukey(r=x, k = 1.5)
```

---

residuals.backf   *Residuals for objects of class* backf

Description

This function returns the residuals of the fitted additive model using the classical or robust backfitting estimators, as computed with backf.cl or backf.rob.

Usage

```
## S3 method for class 'backf'
residuals(object, ...)
```

Arguments

object	an object of class backf, a result of a call to backf.cl or backf.rob.
...	additional other arguments. Currently ignored.

Value

A vector of residuals.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

---

summary.backf   *Summary for additive models fits using backfitting*

Description

Summary method for class backf.

Usage

```
## S3 method for class 'backf'
summary(object, ...)
```

Arguments

object	an object of class backf, a result of a call to backf.cl or backf.rob.
...	additional other arguments. Currently ignored.

Details

This function returns the estimation of the intercept and also the five-number summary and the mean of the residuals for both classical and robust estimators. For the robust estimator it also returns the estimate of the residual standard error.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

Index