

Package ‘ibr’

March 5, 2012

Version 1.4.1

Date 05-03-2012

Title Iterative Bias Reduction

Author Pierre-Andre Cornillon, Nicolas Hengartner, Eric Matzner-Lober

Maintainer Pierre-Andre Cornillon <pierre-andre.cornillon@univ-rennes2.fr>

Imports nlme, graphics, stats

Description an R package for multivariate smoothing

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-03-05 19:06:02

R topics documented:

ibr-package	2
AIC.ibr	3
betaA	4
betaS1	5
BIC	6
bwchoice	8
calcA	9
cvobs	10
departnoyau	11
dssmoothe	12
dsSx	13
DuchonQ	14
DuchonS	15
fittedA	15
fittedS1	16
forward	17

ibr	20
iterchoiceA	23
iterchoiceAcv	25
iterchoiceAcve	26
iterchoiceAe	28
iterchoiceS1	29
iterchoiceS1cv	30
iterchoiceS1cve	31
iterchoiceS1e	33
kernel	34
kernelSx	35
lambdchoice	35
npregression	36
ozone	38
plot.forwardibr	39
plot.ibr	40
poids	41
predict.ibr	42
predict.npregression	43
print.summary.ibr	44
print.summary.npregression	45
summary.ibr	46
summary.npregression	47
sumvalpr	48
tracekernel	49
Index	50

ibr-package

*Iterative Bias Reduction***Description**

an R package for multivariate smoothing using Iterative Bias Reduction smoother.

Details

- We are interested in smoothing (the values of) a vector of n observations y by d covariates measured at the same n observations (gathered in the matrix X). The iterated Bias Reduction produces a sequence of smoothers

$$\hat{y} = S_k y = (I - (I - S)^k) y,$$

where S is the pilot smoother which can be either a kernel or a thin plate spline smoother. In case of a kernel smoother, the kernel is built as a product of univariate kernels.

- The most important parameter of the iterated bias reduction is k the number of iterations. Usually this parameter is unknown and is chosen from the search grid K to minimize the criterion (GCV, AIC, AICc, BIC or gMDL).
The user must choose the pilot smoother (kernel "k" or thin plate splines "tps") plus the values of bandwidths (kernel) or λ thin plate splines). As the choice of these raw values depend on each particular dataset, one can rely on effective degrees of freedom or default values given as degree of freedom, see argument `df` of the main function `ibr`.

Index of functions to be used by end user:

```
ibr:                Iterative bias reduction smoothing
plot.ibr:           Plot diagnostic for an ibr object
predict.ibr:        Predicted values using iterative bias reduction
                    smoothers
print.summary.ibr: Printing iterative bias reduction summaries
summary.ibr:        Summarizing iterative bias reduction fits
```

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner, Eric Matzner-Lober

Maintainer: Pierre-Andre Cornillon <pierre-andre.cornillon@supagro.inra.fr>

Examples

```
## Not run:
data(ozone, package = "ibr")
res.ibr <- ibr(ozone[,-1],ozone[,1],smoother="k",df=1.1)
summary(res.ibr)
predict(res.ibr)
plot(res.ibr)

## End(Not run)
```

AIC.ibr

Summarizing iterative bias reduction fits

Description

Generic function calculating the Akaike information criterion for one model objects of `ibr` class for which a log-likelihood value can be obtained, according to the formula $-2 \log(\sigma^2) + kdf/n$, where df represents the effective degree of freedom (trace) of the smoother in the fitted model, and $k = 2$ for the usual AIC, or $k = \log(n)$ (n the number of observations) for the so-called BIC or SBC (Schwarz's Bayesian criterion).

Usage

```
## S3 method for class 'ibr'
AIC(object, ..., k = 2)
```

Arguments

object	A fitted model object of class <code>ibr</code> .
...	Not used.
k	Numeric, the <i>penalty</i> per parameter to be used; the default <code>k = 2</code> is the classical AIC.

Details

The `ibr` method for AIC, `AIC.ibr()` calculates $\log(\sigma^2) + 2 * df/n$, where df is the trace of the smoother.

Value

returns a numeric value with the corresponding AIC (or BIC, or ..., depending on `k`).

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Hurvich, C. M., Simonoff J. S. and Tsai, C. L. (1998) Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion. *Journal of the Royal Statistical Society, Series B*, 60, 271-293 .

See Also

[ibr](#), [summary.ibr](#)

Examples

```
## Not run: data(ozone, package = "ibr")
res.ibr <- ibr(ozone[,-1],ozone[,1],df=1.2)
summary(res.ibr)
predict(res.ibr)
## End(Not run)
```

betaA

Calculates coefficients for iterative bias reduction smoothers

Description

Calculates the coefficients for the iterative bias reduction smoothers. This function is not intended to be used directly.

Usage

```
betaA(n, eigenvaluesA, tPADmdemiY, DdemiPA, ddlmini, k, index0)
```

Arguments

n	The number of observations.
eigenvaluesA	Vector of the eigenvalues of the symmetric matrix A .
tPADmdemiY	The transpose of the matrix of eigen vectors of the symmetric matrix A times the inverse of the square root of the diagonal matrix D .
DdemiPA	The square root of the diagonal matrix D times the eigen vectors of the symmetric matrix A .
ddlmini	The number of eigenvalues (numerically) equals to 1.
k	A scalar which gives the number of iterations.
index0	The index of the first eigen values of S numerically equal to 0.

Details

See the reference for detailed explanation of A and D and the meaning of coefficients.

Value

Returns the vector of coefficients (of length n , the number of observations.)

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#)

betaS1

Coefficients for iterative bias reduction method.

Description

The function evaluates the smoothing matrix H , the matrices Q and S and their associated coefficients c and s . This function is not intended to be used directly.

Usage

```
betaS1(n,U,tUy,eigenvaluesS1,ddlmini,k,lambda,Sgu,Qgu,index0)
```

Arguments

n	The number of observations.
U	The the matrix of eigen vectors of the symmetric smoothing matrix S .
tUy	The transpose of the matrix of eigen vectors of the symmetric smoothing matrix S times the vector of observation y .
eigenvaluesS1	Vector of the eigenvalues of the symmetric smoothing matrix S .
ddlmini	The number of eigen values of S equal to 1.
k	A numeric vector which give the number of iterations.
lambda	The smoothness coefficient lambda for thin plate splines of order m .
Sgu	The matrix of the polynomial null space S .
Qgu	The matrix of the semi kernel (or radial basis) Q .
index0	The index of the first eigen values of S numerically equal to 0.

Details

See the reference for detailed explanation of Q (the semi kernel or radial basis) and S (the polynomial null space).

Value

Returns a list containing of coefficients for the null space dgub and the semi-kernel cgub

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober

References

C. Gu (2002) *Smoothing spline anova models*. New York: Springer-Verlag.

See Also

[ibr](#)

BIC

Information Criterion for ibr

Description

Functions calculating the Bayesian Informative Criterion , the Generalized Cross Validation criterion and the Corrected Akaike information criterion.

Usage

```
## S3 method for class 'ibr'
BIC(object, ...)

## S3 method for class 'ibr'
GCV(object, ...)

## S3 method for class 'ibr'
AICc(object, ...)
```

Arguments

```
object      A fitted model object of class ibr.
...         Only for compatibility purpose with BIC of nlme package.
```

Details

The ibr method for BIC, `BIC.ibr()` calculates $\log(\sigma^2) + \log(n) * df/n$, where df is the trace of the smoother.

The ibr method for GCV, `GCV.ibr()` calculates $\log(\sigma^2) - 2 * \log(1 - df/n)$

The ibr method for AICc, `AICc.ibr()` calculates $\log(\sigma^2) + 1 + (2 * (df + 1))/(n - df - 2)$.

Value

Returns a numeric value with the corresponding BIC, GCV or AICc.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Hurvich, C. M., Simonoff J. S. and Tsai, C. L. (1998) Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion. *Journal of the Royal Statistical Society, Series B*, 60, 271-293 .

See Also

[ibr](#), [summary.ibr](#)

Examples

```
## Not run: data(ozone, package = "ibr")
res.ibr <- ibr(ozone[,-1],ozone[,1])
BIC(res.ibr)
GCV(res.ibr)
AICc(res.ibr)

## End(Not run)
```

bwchoice	<i>Choice of bandwidth achieving a prescribed effective degree of freedom</i>
----------	---

Description

Perform a search for the bandwidths in the given grid. For each explanatory variable, the bandwidth is chosen such that the trace of the smoothing matrix according to that variable (effective degree of freedom) is equal to a prescribed value. This function is not intended to be used directly.

Usage

```
bwchoice(X,objectif,kernelx="g",itermax=1000)
```

Arguments

X	A matrix with n rows (individuals) and p columns (numeric variables).
objectif	A numeric vector of either length 1 or length equal to the number of columns of X. It indicates the desired effective degree of freedom (trace) of the smoothing matrix for each variable. <code>objectif</code> is repeated when the length of vector <code>objectif</code> is 1.
kernelx	String which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q").
itermax	A scalar which controls the number of iterations for that search.

Value

Returns a vector of length d , the number of explanatory variable, where each coordinate is the value of the selected bandwidth for each explanatory variable

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

See Also

[ibr](#)

`calcA`*Decomposition of the kernel smoother*

Description

Calculates the decomposition of the kernel smoothing matrix in two part: a diagonal matrix D and a symmetric matrix A . This function is not intended to be used directly.

Usage

```
calcA(X, bx, kernelx="g")
```

Arguments

<code>X</code>	The matrix of explanatory variables, size n, p .
<code>bx</code>	The vector of bandwidth of length p .
<code>kernelx</code>	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q").

Details

see the reference for detailed explanation of A and D and the meaning of coefficients.

Value

Returns a list containing two matrices: the symmetric matrix A in component `A`) and the square root of the diagonal matrix D in the component `Ddemi` and the trace of the smoother in the component `df`.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

See Also

[ibr](#)

cvobs	<i>Selection of the number of iterations for iterative bias reduction smoothers</i>
-------	---

Description

The function `cvobs` gives the index of observations in each test set. This function is not intended to be used directly.

Usage

```
cvobs(n, ntest, ntrain, Kfold, type=
c("random", "timeseries", "consecutive", "interleaved"), npermut, seed)
```

Arguments

<code>n</code>	The total number of observations.
<code>ntest</code>	The number of observations in test set.
<code>ntrain</code>	The number of observations in training set.
<code>Kfold</code>	Either the number of folds or a boolean or NULL.
<code>type</code>	A character string in <code>random</code> , <code>timeseries</code> , <code>consecutive</code> , <code>interleaved</code> and give the type of segments.
<code>npermut</code>	The number of random draw (with replacement), used for <code>type="random"</code> .
<code>seed</code>	Controls the seed of random generator (via set.seed).

Value

Returns a list with in each component the index of observations to be used as a test set.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#)

`departnoyau`*Trace of the product kernel smoother*

Description

Search bandwidth for each univariate kernel smoother such that the product of these univariate kernel gives a kernel smoother with a chosen effective degree of freedom (trace of the smoother). The bandwidths are constrained to give, for each explanatory variable, a kernel smoother with same trace as the others. This function is not intended to be used directly.

Usage

```
departnoyau(df, x, kernel, dftobwitmax, n, p, dfobjectif)
```

Arguments

<code>df</code>	A numeric vector giving the effective degree of freedom (trace) of the univariate smoothing matrix for each variable of x .
<code>x</code>	Matrix of explanatory variables, size n, p .
<code>kernel</code>	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q").
<code>dftobwitmax</code>	Specifies the maximum number of iterations transmitted to uniroot function.
<code>n</code>	Number of rows of data matrix x .
<code>p</code>	Number of columns of data matrix x .
<code>dfobjectif</code>	A numeric vector of length 1 which indicates the desired effective degree of freedom (trace) of the smoothing matrix (product kernel smoother) for x .

Value

Returns the desired bandwidths.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

See Also

[ibr](#)

dssmoother	<i>Evaluate the smoothing matrix, the radial basis matrix, the polynomial matrix and their associated coefficients</i>
------------	--

Description

The function evaluates the smoothing matrix H , the matrices Q and S and their associated coefficients c and s . This function is not intended to be used directly.

Usage

```
dssmoother(X, Y=NULL, lambda, m, s)
```

Arguments

X	Matrix of explanatory variables, size n, p .
Y	Vector of response variable. If null, only the smoothing matrix is returned.
λ	The smoothness coefficient λ for thin plate splines of order m .
m	The order of derivatives for the penalty (for thin plate splines it is the order). This integer m must verify $2m+2s/d > 1$, where d is the number of explanatory variables.
s	The power of weighting function. For thin plate splines s is equal to 0. This real must be strictly smaller than $d/2$ (where d is the number of explanatory variables) and must verify $2m+2s/d$. To get pseudo-cubic splines, choose $m=2$ and $s=(d-1)/2$ (See Duchon, 1977).

Details

see the reference for detailed explanation of Q (the semi kernel or radial basis) and S (the polynomial null space).

Value

Returns a list containing the smoothing matrix H , and two matrices denoted S_{gu} (for null space) and Q_{gu} .

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober

References

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. in W. Schemp and K. Zeller (eds) *Construction theory of functions of several variables*, 85-100, Springer, Berlin.

C. Gu (2002) *Smoothing spline anova models*. New York: Springer-Verlag.

See Also[ibr](#)

dsSx

*Evaluate the smoothing matrix at any point***Description**

The function evaluates the matrix Q and S related to the explanatory variables X at any points. This function is not intended to be used directly.

Usage

```
dsSx(X,Xetoile,m=2,s=0)
```

Arguments

X	Matrix of explanatory variables, size n,p.
Xetoile	Matrix of new observations with the same number of variables as X , size m,p.
m	The order of derivatives for the penalty (for thin plate splines it is the order). This integer m must verify $2m+2s/d > 1$, where d is the number of explanatory variables.
s	The power of weighting function. For thin plate splines s is equal to 0. This real must be strictly smaller than $d/2$ (where d is the number of explanatory variables) and must verify $2m+2s/d$. To get pseudo-cubic splines, choose $m=2$ and $s=(d-1)/2$ (See Duchon, 1977).

Details

see the reference for detailed explanation of Q (the semi kernel) and S (the polynomial null space).

Value

Returns a list containing two matrices denoted Sgu (for null space) and Qgu

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober

References

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Solobev spaces. in W. Shemp and K. Zeller (eds) *Construction theory of functions of several variables*, 85-100, Springer, Berlin.

C. Gu (2002) *Smoothing spline anova models*. New York: Springer-Verlag.

See Also[ibr](#)

DuchonQ*Computes the semi-kernel of Duchon splines*

Description

The function `DuchonQ` computes the semi-kernel of Duchon splines. This function is not intended to be used directly.

Usage

```
DuchonQ(x, xk, m=2, s=0, symmetric=TRUE)
```

Arguments

<code>x</code>	A numeric matrix of explanatory variables, with n rows and p columns.
<code>xk</code>	A numeric matrix of explanatory variables, with nk rows and p columns.
<code>m</code>	Order of derivatives.
<code>s</code>	Exponent for the weight function.
<code>symmetric</code>	Boolean: if TRUE only <code>x</code> is used and it computes the semi-kernel at observations of <code>x</code> (it should give the same result as <code>DuchonQ(x, xk, m, s, FALSE)</code>).

Value

The semi-kernel evaluated.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. in W. Schempp and K. Zeller (eds) *Construction theory of functions of several variables*, 85-100, Springer, Berlin.

See Also[ibr](#)

DuchonS	<i>Computes the semi-kernel of Duchon splines</i>
---------	---

Description

The function DuchonS computes the semi-kernel of Duchon splines. This function is not intended to be used directly.

Usage

```
DuchonS(x, m=2)
```

Arguments

x	A numeric matrix of explanatory variables, with n rows and p columns.
m	Order of derivatives.

Value

The polynomial part evaluated.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. in W. Schempp and K. Zeller (eds) *Construction theory of functions of several variables*, 85-100, Springer, Berlin.

See Also

[ibr](#)

fittedA	<i>Evaluates the fits for iterative bias reduction method</i>
---------	---

Description

Evaluates the fits for the iterative bias reduction smoother, using a kernel smoother and its decomposition into a symmetric matrix and a diagonal matrix. This function is not intended to be used directly.

Usage

```
fittedA(n, eigenvaluesA, tPADmdemiY, DdemiPA, ddlmini, k)
```

Arguments

n	The number of observations.
eigenvaluesA	Vector of the eigenvalues of the symmetric matrix A .
tPADmdemiY	The transpose of the matrix of eigen vectors of the symmetric matrix A times the inverse of the square root of the diagonal matrix D .
DdemiPA	The square root of the diagonal matrix D times the eigen vectors of the symmetric matrix A .
ddlmini	The number of eigenvalues (numerically) equals to 1.
k	A scalar which gives the number of iterations.

Details

See the reference for detailed explanation of A and D .

Value

Returns a list of two components: `fitted` contains fitted values and `trace` contains the trace (effective degree of freedom) of the iterated bias reduction smoother.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#)

fittedS1

Evaluate the fit for iterative bias reduction model

Description

The function evaluates the fit for iterative bias reduction model for iteration k . This function is not intended to be used directly.

Usage

```
fittedS1(n,U,tUy,eigenvaluesS1,ddlmini,k)
```

Arguments

n	The number of observations.
U	The the matrix of eigen vectors of the symmetric smoothing matrix S .
tUy	The transpose of the matrix of eigen vectors of the symmetric smoothing matrix S times the vector of observation y .
eigenvaluesS1	Vector of the eigenvalues of the symmetric smoothing matrix S .
ddlmini	The number of eigen values of S equal to 1.
k	A numeric vector which gives the number of iterations

Details

see the reference for detailed explanation of computation of iterative bias reduction smoother

Value

Returns a vector containing the fit

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#)

forward

Iterative bias reduction smoothing

Description

Performs a forward variable selection for iterative bias reduction using kernel or thin plate splines. In the latter case, the order m is chosen as the first integer such that $2m/d > 1$, where d is the number of explanatory variables. Missing values are not allowed.

Usage

```
forward(X, Y, criterion="gcv", df=1.5, Kmin=1, Kmax=10000, smoother="k", kernel="g",
  control.par=list(), cv.options=list(), varcrit=criterion)
```

Arguments

<code>X</code>	A numeric matrix of explanatory variables, with n rows and p columns.
<code>Y</code>	A numeric vector of variable to be explained of length n .
<code>criterion</code>	Character string. If the number of iterations (<code>iter</code>) is missing or NULL the number of iterations is chosen using <code>criterion</code> . The criteria available are GCV (default, "gcv"), AIC ("aic"), corrected AIC ("aicc"), BIC ("bic"), gMDL ("gmdl"), map ("map") or rmse ("rmse"). The last two are designed for cross-validation.
<code>df</code>	A numeric vector of either length 1 or length equal to the number of columns of <code>x</code> . If <code>smoother="k"</code> , it indicates the desired degree of freedom (trace) of the smoothing matrix for each variable or for the initial smoother (see <code>contr.sp\$df\$total</code>); <code>df</code> is repeated when the length of vector <code>df</code> is 1. If <code>smoother="tps"</code> , the minimum <code>df</code> of thin plate splines is multiplied by <code>df</code> . This argument is useless if <code>bandwidth</code> is supplied (non null).
<code>Kmin</code>	The minimum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
<code>Kmax</code>	The maximum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
<code>smoother</code>	Character string which allows to choose between thine plate splines "tps" or kernel ("k").
<code>kernel</code>	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q"). The default (gaussian kernel) is strongly advised.
<code>control.par</code>	<p>a named list that control optional parameters. The components are <code>bandwidth</code> (default to NULL), <code>iter</code> (default to NULL), <code>really.big</code> (default to FALSE), <code>dftobwitmax</code> (default to 1000), <code>exhaustive</code> (default to FALSE), <code>m</code> (default to NULL), <code>df\$total</code> (default to FALSE), <code>accuracy</code> (default to 0.01), <code>ddlmaxi</code> (default to $2n/3$) and <code>fraction</code> (default to $c(100, 200, 500, 1000, 5000, 10^4, 5e+04, 1e+05, 5e+05, 1e+06)$).</p> <p><code>bandwidth</code>: a vector of either length 1 or length equal to the number of columns of <code>x</code>. If <code>smoother="k"</code>, it indicates the bandwidth used for each variable, <code>bandwidth</code> is repeated when the length of vector <code>bandwidth</code> is 1. If <code>smoother="tps"</code>, it indicates the amount of penalty (coefficient <code>lambda</code>). The default (missing) indicates, for <code>smoother="k"</code>, that <code>bandwidth</code> for each variable is chosen such that each univariate kernel smoother (for each explanatory variable) has <code>df</code> degrees of freedom and for <code>smoother="tps"</code> that <code>lambda</code> is chosen such that the <code>df</code> of the smoothing matrix is <code>df</code> times the minimum <code>df</code>.</p> <p><code>iter</code>: the number of iterations. If null or missing, an optimal number of iterations is chosen from the search grid (integer from <code>Kmin</code> to <code>Kmax</code>) to minimize the <code>criterion</code>.</p> <p><code>really.big</code>: a boolean: if TRUE it overrides the limitation at 500 observations. Expect long computation times if TRUE.</p> <p><code>dftobwitmax</code>: When <code>bandwidth</code> is chosen by specifying the degree of freedom (see <code>df</code>) a search is done by <code>uniroot</code>. This argument specifies the maximum number of iterations transmitted to <code>uniroot</code> function.</p>

	<p>exhaustive: boolean, if TRUE an exhaustive search of optimal number of iteration on the grid $K_{min}:K_{max}$ is performed. If FALSE the minimum of criterion is searched using <code>optimize</code> between K_{min} and K_{max}.</p> <p>m: the order of thin plate splines. This integer m must verifies $2m/d > 1$, where d is the number of explanatory variables. The missing default to choose the order m as the first integer such that $2m/d > 1$, where d is the number of explanatory variables (same for NULL).</p> <p><code>dftotal</code>: a boolean wich indicates when FALSE that the argument <code>df</code> is the objective <code>df</code> for each univariate kernel (the default) calculated for each explanatory variable or for the overall (product) kernel, that is the base smoother (when TRUE).</p> <p><code>accuracy</code>: tolerance when searching bandwidths which lead to a chosen overall intial <code>df</code>.</p> <p><code>dfmaxi</code>: the maximum degree of freedom allowed for iterated biased reduction smoother.</p> <p><code>fraction</code>: the subdivision of interval K_{min},K_{max} if non exhaustive search is performed (see also <code>iterchoiceA</code> or <code>iterchoiceS1</code>).</p>
<code>cv.options</code>	<p>A named list which controls the way to do cross validation with component <code>bwchange</code>, <code>nptest</code>, <code>ntrain</code>, <code>Kfold</code>, <code>type</code>, <code>seed</code>, <code>method</code> and <code>npermut</code>. <code>bwchange</code> is a boolean (default to FALSE) which indicates if bandwidth have to be recomputed each time. <code>nptest</code> is the number of observations in test set and <code>ntrain</code> is the number of observations in training set. Actually, only one of these is needed the other can be NULL or missing. <code>Kfold</code> a boolean or an integer. If <code>Kfold</code> is TRUE then the number of fold is deduced from <code>nptest</code> (or <code>ntrain</code>). <code>type</code> is a character string in <code>random</code>, <code>timeseries</code>, <code>consecutive</code>, <code>interleaved</code> and give the type of segments. <code>seed</code> controls the seed of random generator. <code>method</code> is either "inmemory" or "outmemory"; "inmemory" induces some calculations outside the loop saving computational time but leading to an increase of the required memory. <code>npermut</code> is the number of random draws. If <code>cv.options</code> is <code>list()</code>, then component <code>nptest</code> is set to <code>floor(nrow(x)/10)</code>, <code>type</code> is <code>random</code>, <code>npermut</code> is 20 and <code>method</code> is "inmemory", and the other components are NULL.</p>
<code>varcrit</code>	<p>Character string. Criterion used for variable selection. The criteria available are GCV, AIC ("aic"), corrected AIC ("aicc"), BIC ("bic") and gMDL ("gmdl").</p>

Value

Returns an object of class `forwardibr` which is a matrix with p columns. In the first row, each entry j contains the value of the chosen criterion for the univariate smoother using the j th explanatory variable. The variable which realize the minimum of the first row is included in the model. All the column of this variable will be `Inf` except the first row. In the second row, each entry j contains the bivariate smoother using the j th explanatory variable and the variable already included. The variable which realize the minimum of the second row is included in the model. All the column of this variable will be `Inf` except the two first row. This forward selection process continue until the chosen criterion increases.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [plot.forwardibr](#)

Examples

```
## Not run:
data(ozone, package = "ibr")
res.ibr <- forward(ozone[,-1],ozone[,1],df=1.2)
apply(res.ibr,1,which.min)

## End(Not run)
```

ibr	<i>Iterative bias reduction smoothing</i>
-----	---

Description

Performs iterative bias reduction using kernel, thin plate splines or Duchon splines. Missing values are not allowed.

Usage

```
ibr(x, y, criterion="gcv", df=1.5, Kmin=1, Kmax=1e+06, smoother="k",
    kernel="g", control.par=list(), cv.options=list())
```

Arguments

x	A numeric matrix of explanatory variables, with n rows and p columns.
y	A numeric vector of variable to be explained of length n .
criterion	Character string. If the number of iterations (<code>iter</code>) is missing or NULL the number of iterations is chosen using <code>criterion</code> . The criteria available are GCV (default, "gcv"), AIC ("aic"), corrected AIC ("aicc"), BIC ("bic"), gMDL ("gmdl"), map ("map") or rmse ("rmse"). The last two are designed for cross-validation.
df	A numeric vector of either length 1 or length equal to the number of columns of <code>x</code> . If <code>smoother="k"</code> , it indicates the desired effective degree of freedom (trace) of the smoothing matrix for each variable or for the initial smoother (see <code>contr.sp\$df\$total</code>); <code>df</code> is repeated when the length of vector <code>df</code> is 1. If <code>smoother="tps"</code> or <code>smoother="ds"</code> , the minimum <code>df</code> of splines is multiplied by <code>df</code> . This argument is useless if bandwidth is supplied (non null).
Kmin	The minimum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.

Kmax	The maximum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
smoother	Character string which allows to choose between thin plate splines "tps", Duchon splines "tps" (see Duchon, 1977) or kernel ("k").
kernel	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q"). The default (gaussian kernel) is strongly advised.
control.par	<p>A named list that control optional parameters. The components are bandwidth (default to NULL), iter (default to NULL), really.big (default to FALSE), dftobwitmax (default to 1000), exhaustive (default to FALSE), m (default to NULL), s (default to NULL), dftotal (default to FALSE), accuracy (default to 0.01), ddlmaxi (default to $2n/3$) and fraction (default to c(100, 200, 500, 1000, 5000, 10^4, $5e+04$, $1e+05$, $5e+05$, $1e+06$)).</p> <p>bandwidth: a vector of either length 1 or length equal to the number of columns of x. If smoother="k", it indicates the bandwidth used for each variable, bandwidth is repeated when the length of vector bandwidth is 1. If smoother="tps", it indicates the amount of penalty (coefficient lambda). The default (missing) indicates, for smoother="k", that bandwidth for each variable is chosen such that each univariate kernel smoother (for each explanatory variable) has df effective degrees of freedom and for smoother="tps" or smoother="ds" that lambda is chosen such that the df of the smoothing matrix is df times the minimum df.</p> <p>iter: the number of iterations. If null or missing, an optimal number of iterations is chosen from the search grid (integer from Kmin to Kmax) to minimize the criterion.</p> <p>really.big: a boolean: if TRUE it overrides the limitation at 500 observations. Expect long computation times if TRUE.</p> <p>dftobwitmax: When bandwidth is chosen by specifying the effective degree of freedom (see df) a search is done by uniroot. This argument specifies the maximum number of iterations transmitted to uniroot function.</p> <p>exhaustive: boolean, if TRUE an exhaustive search of optimal number of iteration on the grid Kmin:Kmax is performed. If FALSE the minimum of criterion is searched using optimize between Kmin and Kmax.</p> <p>m: The order of derivatives for the penalty (for thin plate splines it is the order). This integer m must verify $2m+2s/d > 1$, where d is the number of explanatory variables. The default (for smoother="tps") is to choose the order m as the first integer such that $2m/d > 1$, where d is the number of explanatory variables. The default (for smoother="ds") is to choose $m=2$ (p pseudo cubic splines).</p> <p>s: the power of weighting function. For thin plate splines s is equal to 0. This real must be strictly smaller than $d/2$ (where d is the number of explanatory variables) and must verify $2m+2s/d$. To get pseudo-cubic splines (the default), choose $m=2$ and $s=(d-1)/2$ (See Duchon, 1977).the order of thin plate splines. This integer m must verifies $2m/d > 1$, where d is the number of explanatory variables.</p> <p>dftotal: a boolean wich indicates when FALSE that the argument df is the objective df for each univariate kernel (the default) calculated for each explanatory variable or for the overall (product) kernel, that is the base smoother (when TRUE).</p>

	accuracy: tolerance when searching bandwidths which lead to a chosen overall initial df.
	dfmaxi: the maximum effective degree of freedom allowed for iterated biased reduction smoother.
	fraction: the subdivision of interval K_{min}, K_{max} if non exhaustive search is performed (see also iterchoiceA or iterchoiceS1).
	scale: boolean. If TRUE x is scaled (using scale); default to FALSE.
cv.options	A named list which controls the way to do cross validation with component <code>bwchange</code> , <code>ntest</code> , <code>ntrain</code> , <code>Kfold</code> , <code>type</code> , <code>seed</code> , <code>method</code> and <code>npermut</code> . <code>bwchange</code> is a boolean (default to FALSE) which indicates if bandwidth have to be recomputed each time. <code>ntest</code> is the number of observations in test set and <code>ntrain</code> is the number of observations in training set. Actually, only one of these is needed the other can be NULL or missing. <code>Kfold</code> a boolean or an integer. If <code>Kfold</code> is TRUE then the number of fold is deduced from <code>ntest</code> (or <code>ntrain</code>). <code>type</code> is a character string in <code>random</code> , <code>timeseries</code> , <code>consecutive</code> , <code>interleaved</code> and give the type of segments. <code>seed</code> controls the seed of random generator. <code>method</code> is either "inmemory" or "outmemory"; "inmemory" induces some calculations outside the loop saving computational time but leading to an increase of the required memory. <code>npermut</code> is the number of random draws. If <code>cv.options</code> is <code>list()</code> , then component <code>ntest</code> is set to <code>floor(nrow(x)/10)</code> , <code>type</code> is <code>random</code> , <code>npermut</code> is 20 and <code>method</code> is "inmemory", and the other components are NULL

Value

Returns an object of class `ibr` which is a list including:

<code>beta</code>	Vector of coefficients.
<code>residuals</code>	Vector of residuals.
<code>fitted</code>	Vector of fitted values.
<code>iter</code>	The number of iterations used.
<code>initialdf</code>	The initial effective degree of freedom of the pilot (or base) smoother.
<code>finaldf</code>	The effective degree of freedom of the iterated bias reduction smoother at the <code>iter</code> iterations.
<code>bandwidth</code>	Vector of bandwidth for each explanatory variable
<code>call</code>	A list containing several components: <code>x</code> contains the initial explanatory variables (scaled if relevant), <code>y</code> contains the initial dependant variables, <code>criterion</code> contains the chosen criterion, <code>kernel</code> the kernel, <code>p</code> the number of explanatory variables and <code>m</code> the order of the splines (if relevant), <code>s</code> the power of weights, <code>scaled</code> boolean which is TRUE when explanatory variables are scaled, <code>mean</code> mean of explanatory variables if <code>scaled=TRUE</code> , <code>sd</code> standard deviation of explanatory variables if <code>scaled=TRUE</code> .
<code>criteria</code>	either a list containing all the criteria evaluated on the grid $K_{min}:K_{max}$ (along with the effective degree of freedom of the smoother and the sigma squared on this grid) if an exhaustive search is chosen (see the value of function iterchoiceAe or iterchoiceS1e) or the value of the chosen criterion at the given iteration if a non exhaustive search is chosen (see <code>exhaustive</code>). If the number of iterations <code>iter</code> is given by the user NULL is returned

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[predict.ibr](#), [summary.ibr](#)

Examples

```
f <- function(x, y) { .75*exp(-((9*x-2)^2 + (9*y-2)^2)/4) +
  .75*exp(-((9*x+1)^2/49 + (9*y+1)^2/10)) +
  .50*exp(-((9*x-7)^2 + (9*y-3)^2)/4) -
  .20*exp(-((9*x-4)^2 + (9*y-7)^2)) }
# define a (fine) x-y grid and calculate the function values on the grid
ngrid <- 50; xf <- seq(0,1, length=ngrid+2)[-c(1,ngrid+2)]
yf <- xf ; zf <- outer(xf, yf, f)
grid <- cbind(rep(xf, ngrid), rep(xf, rep(ngrid, ngrid)))
persp(xf, yf, zf, theta=130, phi=20, expand=0.45,main="True Function")
#generate a data set with function f and noise to signal ratio 5
noise <- .2 ; N <- 100
xr <- seq(0.05,0.95,by=0.1) ; yr <- xr ; zr <- outer(xr,yr,f) ; set.seed(25)
std <- sqrt(noise*var(as.vector(zr))) ; noise <- rnorm(length(zr),0,std)
Z <- zr + matrix(noise,sqrt(N),sqrt(N))
# transpose the data to a column format
xc <- rep(xr, sqrt(N)) ; yc <- rep(yr, rep(sqrt(N),sqrt(N)))
X <- cbind(xc, yc) ; Zc <- as.vector(Z)
# fit by thin plate splines (of order 2) ibr
res.ibr <- ibr(X,Zc,df=1.1,smoother="tps")
fit <- matrix(predict(res.ibr,grid),ngrid,ngrid)
persp(xf, yf, fit ,theta=130,phi=20,expand=0.45,main="Fit",zlab="fit")

## Not run:
data(ozone, package = "ibr")
res.ibr <- ibr(ozone[,-1],ozone[,1],df=1.1)
summary(res.ibr)
predict(res.ibr)
## End(Not run)
```

Description

The function `iterchoiceA` searches the interval from `mini` to `maxi` for a minimum of the function which calculates the chosen criterion (`critAgcv`, `critAaic`, `critAbic`, `critAaicc` or `critAgmdl`) with respect to its first argument (a given iteration `k`) using `optimize`. This function is not intended to be used directly.

Usage

```
iterchoiceA(n, mini, maxi, eigenvaluesA, tPADmdemiY, DdemiPA,
ddlmini, ddlmaxi, y, criterion, fraction)
```

Arguments

<code>n</code>	The number of observations.
<code>mini</code>	The lower end point of the interval to be searched.
<code>maxi</code>	The upper end point of the interval to be searched.
<code>eigenvaluesA</code>	Vector of the eigenvalues of the symmetric matrix A .
<code>tPADmdemiY</code>	The transpose of the matrix of eigen vectors of the symmetric matrix A times the inverse of the square root of the diagonal matrix D .
<code>DdemiPA</code>	The square root of the diagonal matrix D times the eigen vectors of the symmetric matrix A .
<code>ddlmini</code>	The number of eigenvalues (numerically) equals to 1.
<code>ddlmaxi</code>	The maximum df. No criterion is calculated and <code>Inf</code> is returned.
<code>y</code>	The vector of observations of dependant variable.
<code>criterion</code>	The criteria available are GCV (default, "gcv"), AIC ("aic"), corrected AIC ("aicc"), BIC ("bic") or gMDL ("gmdl").
<code>fraction</code>	The subdivision of the interval <code>[mini,maxi]</code> .

Details

See the reference for detailed explanation of A and D . The interval `[mini,maxi]` is splitted into subintervals using `fraction`. In each subinterval the function `fcriterion` is minimized using `optimize` (with respect to its first argument) and the minimum (and its argument) of the result of these optimizations is returned.

Value

A list with components `iter` and `objective` which give the (rounded) optimum number of iterations (between `Kmin` and `Kmax`) and the value of the function at that real point (not rounded).

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [iterchoiceA](#)

iterchoiceAcv	<i>Selection of the number of iterations for iterative bias reduction smoothers</i>
---------------	---

Description

The function `iterchoiceAcv` searches the interval from `mini` to `maxi` for a minimum of the function criterion with respect to its first argument using `optimize`. This function is not intended to be used directly.

Usage

```
iterchoiceAcv(X, y, bx, df, kernelx, ddlmini, ntest, ntrain, Kfold,
type, npermut, seed, Kmin, Kmax, criterion, fraction)
```

Arguments

<code>X</code>	A numeric matrix of explanatory variables, with n rows and p columns.
<code>y</code>	A numeric vector of variable to be explained of length n .
<code>bx</code>	The vector of different bandwidths, length p .
<code>df</code>	A numeric vector of either length 1 or length equal to the number of columns of <code>x</code> . If <code>smoother="k"</code> , it indicates the desired effective degree of freedom (trace) of the smoothing matrix for each variable ; <code>df</code> is repeated when the length of vector <code>df</code> is 1. This argument is useless if <code>bandwidth</code> is supplied (non null).
<code>kernelx</code>	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q"). The default (gaussian kernel) is strongly advised.
<code>ddlmini</code>	The number of eigenvalues (numerically) equals to 1.
<code>ntest</code>	The number of observations in test set.
<code>ntrain</code>	The number of observations in training set.
<code>Kfold</code>	Either the number of folds or a boolean or NULL.
<code>type</code>	A character string in <code>random</code> , <code>timeseries</code> , <code>consecutive</code> , <code>interleaved</code> and give the type of segments.
<code>npermut</code>	The number of random draw (with replacement), used for <code>type="random"</code> .
<code>seed</code>	Controls the seed of random generator (via <code>set.seed</code>).

Kmin	The minimum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
Kmax	The maximum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
criterion	The criteria available are map ("map") or rmse ("rmse").
fraction	The subdivision of the interval [Kmin,Kmax].

Value

Returns the optimum number of iterations (between Kmin and Kmax).

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#)

iterchoiceAcve	<i>Selection of the number of iterations for iterative bias reduction smoothers</i>
----------------	---

Description

Evaluates at each iteration proposed in the grid the cross-validated root mean squared error (RMSE) and mean of the relative absolute error (MAP). The minimum of these criteria gives an estimate of the optimal number of iterations. This function is not intended to be used directly.

Usage

```
iterchoiceAcve(X, y, bx, df, kernelx, ddlmini, ntest, ntrain,
Kfold, type, npermut, seed, Kmin, Kmax)
```

Arguments

X	A numeric matrix of explanatory variables, with n rows and p columns.
y	A numeric vector of variable to be explained of length n .
bx	The vector of different bandwidths, length p .

df	A numeric vector of either length 1 or length equal to the number of columns of x . If smoother="k", it indicates the desired effective degree of freedom (trace) of the smoothing matrix for each variable ; df is repeated when the length of vector df is 1. This argument is useless if bandwidth is supplied (non null).
kernelx	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q"). The default (gaussian kernel) is strongly advised.
ddlmini	The number of eigenvalues (numerically) equals to 1.
ntest	The number of observations in test set.
ntrain	The number of observations in training set.
Kfold	Either the number of folds or a boolean or NULL.
type	A character string in random,timeseries,consecutive, interleaved and give the type of segments.
npermut	The number of random draw (with replacement), used for type="random".
seed	Controls the seed of random generator (via set.seed).
Kmin	The minimum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
Kmax	The maximum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.

Value

Returns the values of RMSE and MAP for each value of the grid K . Inf are returned if the iteration leads to a smoother with a df bigger than `ddlmaxi`.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#)

iterchoiceAe	<i>Selection of the number of iterations for iterative bias reduction smoothers</i>
--------------	---

Description

Evaluates at each iteration proposed in the grid the value of different criteria: GCV, AIC, corrected AIC, BIC and gMDL (along with the ddl and sigma squared). The minimum of these criteria gives an estimate of the optimal number of iterations. This function is not intended to be used directly.

Usage

```
iterchoiceAe(Y, K, eigenvaluesA, tPADmdemiY, DdemiPA, ddlmini,
ddlmaxi)
```

Arguments

Y	The response variable.
K	A numeric vector which give the search grid for iterations.
eigenvaluesA	Vector of the eigenvalues of the symmetric matrix A .
tPADmdemiY	The transpose of the matrix of eigen vectors of the symmetric matrix A times the inverse of the square root of the diagonal matrix D .
DdemiPA	The square root of the diagonal matrix D times the eigen vectors of the symmetric matrix A .
ddlmini	The number of eigenvalues (numerically) which are equal to 1.
ddlmaxi	The maximum df. No criteria are calculated beyond the number of iterations that leads to df bigger than this bound.

Details

See the reference for detailed explanation of A and D

Value

Returns the values of GCV, AIC, corrected AIC, BIC, gMDL, df and sigma squared for each value of the grid K . Inf are returned if the iteration leads to a smoother with a df bigger than `ddlmaxi`.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [iterchoiceA](#)

iterchoiceS1

Number of iterations selection for iterative bias reduction model

Description

The function `iterchoiceS1` searches the interval from `mini` to `maxi` for a minimum of the function which calculates the chosen criterion (`critS1gcv`, `critS1aic`, `critS1bic`, `critS1aicc` or `critS1gmdl`) with respect to its first argument (a given iteration `k`) using `optimize`. This function is not intended to be used directly.

Usage

```
iterchoiceS1(n, mini, maxi, tUy, eigenvaluesS1, ddlmini, ddlmaxi,
y, criterion, fraction)
```

Arguments

<code>n</code>	The number of observations.
<code>mini</code>	The lower end point of the interval to be searched.
<code>maxi</code>	The upper end point of the interval to be searched.
<code>eigenvaluesS1</code>	Vector of the eigenvalues of the symmetric smoothing matrix S .
<code>tUy</code>	The transpose of the matrix of eigen vectors of the symmetric smoothing matrix S times the vector of observation y .
<code>ddlmini</code>	The number of eigen values of S equal to 1.
<code>ddlmaxi</code>	The maximum df. No criterion is calculated and <code>Inf</code> is returned.
<code>y</code>	The vector of observations of dependant variable.
<code>criterion</code>	The criteria available are GCV (default, "gcv"), AIC ("aic"), corrected AIC ("aicc"), BIC ("bic") or gMDL ("gmdl").
<code>fraction</code>	The subdivision of the interval <code>[mini,maxi]</code> .

Details

The interval `[mini,maxi]` is splitted into subintervals using `fraction`. In each subinterval the function `fcriterion` is minimized using `optimize` (with respect to its first argument) and the minimum (and its argument) of the result of these optimizations is returned.

Value

A list with components `iter` and `objective` which give the (rounded) optimum number of iterations (between `Kmin` and `Kmax`) and the value of the function at that real point (not rounded).

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [iterchoiceS1](#)

iterchoiceS1cv	<i>Selection of the number of iterations for iterative bias reduction smoothers with base thin-plate splines or duchon splines smoother</i>
----------------	---

Description

The function `iterchoiceS1cv` searches the interval from `mini` to `maxi` for a minimum of the function criterion with respect to its first argument using `optimize`. This function is not intended to be used directly.

Usage

```
iterchoiceS1cv(X, y, lambda, df, ddlmini, ntest, ntrain,
Kfold, type, npermut, seed, Kmin, Kmax, criterion, m, s,
fraction)
```

Arguments

<code>X</code>	A numeric matrix of explanatory variables, with n rows and p columns.
<code>y</code>	A numeric vector of variable to be explained of length n .
<code>lambda</code>	A numeric positive coefficient that governs the amount of penalty (coefficient <code>lambda</code>).
<code>df</code>	A numeric vector of length 1 which is multiplied by the minimum <code>df</code> of thin plate splines ; This argument is useless if <code>lambda</code> is supplied (non null).
<code>ddlmini</code>	The number of eigenvalues equals to 1.
<code>ntest</code>	The number of observations in test set.
<code>ntrain</code>	The number of observations in training set.
<code>Kfold</code>	Either the number of folds or a boolean or NULL.
<code>type</code>	A character string in <code>random</code> , <code>timeseries</code> , <code>consecutive</code> , <code>interleaved</code> and give the type of segments.
<code>npermut</code>	The number of random draw (with replacement), used for <code>type="random"</code> .
<code>seed</code>	Controls the seed of random generator (via <code>set.seed</code>).

Kmin	The minimum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
Kmax	The maximum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
criterion	The criteria available are map ("map") or rmse ("rmse").
m	The order of derivatives for the penalty (for thin plate splines it is the order). This integer m must verify $2m+2s/d > 1$, where d is the number of explanatory variables.
s	The power of weighting function. For thin plate splines s is equal to 0. This real must be strictly smaller than $d/2$ (where d is the number of explanatory variables) and must verify $2m+2s/d$. To get pseudo-cubic splines, choose $m=2$ and $s=(d-1)/2$ (See Duchon, 1977).
fraction	The subdivision of the interval [Kmin,Kmax].

Value

Returns the optimum number of iterations (between Kmin and Kmax).

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Solobev spaces. in W. Schempp and K. Zeller (eds) *Construction theory of functions of several variables*, 85-100, Springer, Berlin.

See Also

[ibr](#)

iterchoiceS1cve	<i>Selection of the number of iterations for iterative bias reduction smoothers with base thin-plate splines smoother or duchon splines smoother</i>
-----------------	--

Description

Evaluates at each iteration proposed in the grid the cross-validated root mean squared error (RMSE) and mean of the relative absolute error (MAP). The minimum of these criteria gives an estimate of the optimal number of iterations. This function is not intended to be used directly.

Usage

```
iterchoiceS1cve(X, y, lambda, df, ddlmini, ntest, ntrain,
Kfold, type, npermut, seed, Kmin, Kmax, m, s)
```

Arguments

X	A numeric matrix of explanatory variables, with n rows and p columns.
y	A numeric vector of variable to be explained of length n .
lambda	A numeric positive coefficient that governs the amount of penalty (coefficient lambda).
df	A numeric vector of length 1 which is multiplied by the minimum df of thin plate splines ; This argument is useless if lambda is supplied (non null).
ddlmini	The number of eigenvalues equals to 1.
ntest	The number of observations in test set.
ntrain	The number of observations in training set.
Kfold	Either the number of folds or a boolean or NULL.
type	A character string in random,timeseries,consecutive, interleaved and give the type of segments.
npermut	The number of random draw (with replacement), used for type="random".
seed	Controls the seed of random generator (via set.seed).
Kmin	The minimum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
Kmax	The maximum number of bias correction iterations of the search grid considered by the model selection procedure for selecting the optimal number of iterations.
m	The order of derivatives for the penalty (for thin plate splines it is the order). This integer m must verify $2m+2s/d>1$, where d is the number of explanatory variables.
s	The power of weighting function. For thin plate splines s is equal to 0. This real must be strictly smaller than $d/2$ (where d is the number of explanatory variables) and must verify $2m+2s/d$. To get pseudo-cubic splines, choose $m=2$ and $s=(d-1)/2$ (See Duchon).

Value

Returns the values of RMSE and MAP for each value of the grid K. Inf are returned if the iteration leads to a smoother with a df bigger than ddlmaxi.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Solobev spaces. in W. Shemp and K. Zeller (eds) *Construction theory of functions of several variables*, 85-100, Springer, Berlin.

See Also

[ibr](#)

iterchoiceS1e

Number of iterations selection for iterative bias reduction model

Description

Evaluate at each iteration proposed in the grid the value of different criteria: GCV, AIC, corrected AIC, BIC and gMDL (along with the ddl and sigma squared). The minimum of these criteria gives an estimate of the optimal number of iterations. This function is not intended to be used directly.

Usage

```
iterchoiceS1e(y, K, tUy, eigenvaluesS1, ddlmini, ddlmaxi)
```

Arguments

y	The response variable
K	A numeric vector which give the search grid for iterations
eigenvaluesS1	Vector of the eigenvalues of the symmetric smoothing matrix S .
tUy	The transpose of the matrix of eigen vectors of the symmetric smoothing matrix S times the vector of observation y .
ddlmini	The number of eigen values of S equal to 1.
ddlmaxi	The maximum df. No criteria are calculated beyond the number of iterations that leads to df bigger than this bound.

Value

Returns the values of GCV, AIC, corrected AIC, BIC, gMDL, df and sigma squared for each value of the grid K. Inf are returned if the iteration leads to a smoother with a df bigger than ddlmaxi.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [iterchoiceS1](#)

kernel

Kernel evaluation

Description

Evaluate the kernel function at x : Gaussian, Epanechnikov, Uniform, Quartic. This function is not intended to be used directly.

Usage

```
gaussien(X)
epane(X)
uniform(X)
quartic(X)
```

Arguments

X The value where the function has to be evaluate, should be a numeric and can be a scalar, a vector or a matrix

Value

Returns a scalar, a vector or a matrix which coordinates are the values of the kernel at the given coordinate

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

See Also

[ibr](#)

kernelSx	<i>Evaluates the smoothing matrix at x^*</i>
----------	---

Description

The function evaluates the matrix of design weights to predict the response at arbitrary locations x . This function is not intended to be used directly.

Usage

```
kernelSx(kernelx="g",X,Xetoile,bx)
```

Arguments

kernelx	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q").
X	Matrix of explanatory variables, size n, p .
Xetoile	Matrix of new design points x^* at which to predict the response variable, size n^*, p .
bx	The vector of different bandwidths, length p .

Value

Returns the matrix denoted in the paper by Sx, n^*, n .

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

See Also

[ibr](#)

lambdacheice	<i>Choice of bandwidth according to a given effective degree of freedom</i>
--------------	---

Description

Perform a search for the different bandwidths in the given grid. For each explanatory variable, the bandwidth is chosen such that the trace of the smoothing matrix according to that variable (effective degree of freedom) is equal to a given value. This function is not intended to be used directly.

Usage

```
lambdacheice(X,ddlobjectif,m=2,s=0,itermax,smoother="tps")
```

Arguments

<code>X</code>	A matrix with n rows (individuals) and p columns (numeric variables)
<code>ddlobjectif</code>	A numeric vector of length 1 which indicates the desired effective degree of freedom (trace) of the smoothing matrix for thin plate splines of order m .
<code>m</code>	The order of derivatives for the penalty (for thin plate splines it is the order). This integer m must verify $2m+2s/d > 1$, where d is the number of explanatory variables.
<code>s</code>	The power of weighting function. For thin plate splines s is equal to 0. This real must be strictly smaller than $d/2$ (where d is the number of explanatory variables) and must verify $2m+2s/d$. To get pseudo-cubic splines, choose $m=2$ and $s=(d-1)/2$ (See Duchon, 1977).
<code>itermax</code>	A scalar which controls the number of iterations for that search
<code>smoother</code>	Character string which allows to choose between thin plate splines "tps" or Duchon splines "tps" (see Duchon, 1977).

Value

Returns the coefficient lambda that control smoothness for the desired effective degree of freedom

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober

References

Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Solobev spaces. in W. Shemp and K. Zeller (eds) *Construction theory of functions of several variables*, 85-100, Springer, Berlin.

See Also

[ibr](#)

npregression

Local polynomials smoothing

Description

Predicted values from a local polynomials of degree less than 2.
Missing values are not allowed.

Usage

```
npregression(x, y, criterion="rmse", bandwidth=NULL, kernel="g",
             control.par=list(), cv.options=list())
```

Arguments

<code>x</code>	A numeric vector of explanatory variable of length n .
<code>y</code>	A numeric vector of variable to be explained of length n .
<code>criterion</code>	Character string. If the bandwidth (<code>bandwidth</code>) is missing or NULL the number of iterations is chosen using <code>criterion</code> . The criterion available is (cross-validated) rmse (" <code>rmse</code> ") and mean (relative) absolute error.
<code>bandwidth</code>	The kernel bandwidth smoothing parameter (a numeric vector of either length 1).
<code>kernel</code>	Character string which allows to choose between gaussian kernel (" <code>g</code> "), Epanechnikov (" <code>e</code> "), uniform (" <code>u</code> "), quartic (" <code>q</code> ").
<code>control.par</code>	A named list that control optional parameters. The two components are <code>bandwidth</code> for compatibility with <code>ibr</code> arguments and <code>degree</code> which controls the degree of the local polynomial regression. If argument <code>bandwidth</code> is not null or missing, its value is used instead <code>control.par\$bandwidth</code> . <code>degree</code> must be smaller than 2. For (gaussian binned) local polynomial see locpoly
<code>cv.options</code>	A named list which controls the way to do cross validation with component <code>gridbw</code> , <code>ntrain</code> , <code>Kfold</code> , <code>type</code> , <code>seed</code> , <code>method</code> and <code>npermut</code> . <code>gridbw</code> is numeric vector which contains the search grid for optimal bandwidth (default to $1/n * (1+1/n)^{(0:kmax)}$, with $kmax = \text{floor}(\log(n * \text{diff}(\text{range}(x)))/3) / \log(1+1/n)$). <code>ntrain</code> is the number of observations in training set. Actually, only one of these is needed the other can be NULL or missing. <code>Kfold</code> a boolean or an integer. If <code>Kfold</code> is TRUE then the number of fold is deduced from <code>ntrain</code> (or <code>ntrain</code>). <code>type</code> is a character string in <code>random</code> , <code>timeseries</code> , <code>consecutive</code> , <code>interleaved</code> and give the type of segments. <code>seed</code> controls the seed of random generator. <code>npermut</code> is the number of random draws. If <code>cv.options</code> is <code>list()</code> , then component <code>ntrain</code> is set to 1, <code>type</code> is <code>consecutive</code> , <code>Kfold</code> is TRUE, and the other components are NULL, which leads to leave-one-out cross-validation.

Value

Returns an object of class `npregression` which is a list including:

<code>bandwidth</code>	The kernel bandwidth smoothing parameter.
<code>residuals</code>	Vector of residuals.
<code>fitted</code>	Vector of fitted values.
<code>df</code>	The effective degree of freedom of the smoother.
<code>call</code>	A list containing four components: <code>x</code> contains the initial explanatory variables, <code>y</code> contains the initial dependant variables, <code>criterion</code> contains the chosen criterion, <code>kernel</code> the kernel and <code>degree</code> the chosen degree
<code>criteria</code>	either a named list containing the bandwidth search grid and all the criteria (rmse and mae) evaluated on the grid <code>gridbw</code> . If the bandwidth <code>bandwidth</code> is given by the user NULL is returned

Note

See [locpoly](#) for fast binned implementation over an equally-spaced grid of local polynomial. See [ibr](#) for univariate and **multivariate** smoothing.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

See Also

[predict.npregression](#), [summary.npregression](#), [locpoly](#), [ibr](#)

Examples

```
f <- function(x){sin(5*pi*x)}
n <- 100
x <- runif(n)
z <- f(x)
sigma2 <- 0.05*var(z)
erreur <- rnorm(n,0,sqrt(sigma2))
y <- z+erreur
res <- npregression(x,y,bandwidth=0.02)
summary(res)
ord <- order(x)
plot(x,y)
lines(x[ord],predict(res)[ord])
```

ozone

Los Angeles ozone pollution data, 1976.

Description

Los Angeles ozone pollution data, 1976. We deleted from the original data, the first 3 columns which were the Month, Day of the month and Day of the week. Each observation is one day, so there is 366 rows. The ozone data is a matrix with 9 columns.

Format

This data set is a matrix containing the following columns:

[,1]	Ozone	numeric	Daily maximum one-hour-average ozone reading (parts per million) at Upland, CA.
[,2]	Pressure.Vand	numeric	500 millibar pressure height (m) measured at Vandenberg AFB.
[,3]	Wind	numeric	Wind speed (mph) at Los Angeles International Airport (LAX).
[,4]	Humidity	numeric	Humidity in percentage at LAX.

[,5]	Temp.Sand	numeric	Temperature (degrees F) measured at Sandburg, CA.
[,6]	Inv.Base.height	numeric	Inversion base height (feet) at LAX.
[,7]	Pressure.Grad	numeric	Pressure gradient (mm Hg) from LAX to Daggett, CA.
[,8]	Inv.Base.Temp	numeric	Inversion base temperature (degrees F) at LAX.
[,9]	Visibility	numeric	Visibility (miles) measured at LAX.

Source

Leo Breiman, Department of Statistics, UC Berkeley. Data used in Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation, *Journal of American Statistical Association*, **80**, 580–598.

See Also

[ibr](#)

plot.forwardibr	<i>Plot diagnostic for an ibr object</i>
-----------------	--

Description

One plot is currently available: a plot of residuals against fitted values.

Usage

```
## S3 method for class 'forwardibr'
plot(x, global=FALSE, ... )
```

Arguments

x	Object of class forwardibr .
global	Boolean: if global is TRUE the color code is between the min and the max of x (except infinite value); if global is FALSE the color code is between the min and the max of each row.
...	further arguments passed to image .

Value

The function `plot.forwardibr` give an image plot of the values of the criterion obtained by the forward selection process. Image is read from the bottom to the top. At the bottom row, there are all the univariate models and the selected variable is given by the lowest criterion. This variable is selected for the second row. At the second (bottom) row the second variable included is those which give the lowest criterion for this row etc. All the variables included in the final model (selected by forward search) are numbered on the image (by order of inclusion).

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [forward](#)

Examples

```
## Not run: data(ozone, package = "ibr")
ibrsel <- forward(ibr(ozone[,-1],ozone[,1],df=1.2)
plot(ibrsel)
plot(apply(ibrsel,1,min,na.rm=TRUE),type="l")

## End(Not run)
```

plot.ibr

Plot diagnostic for an ibr object

Description

One plot is currently available: a plot of residuals against fitted values.

Usage

```
## S3 method for class 'ibr'
plot(x,... )
```

Arguments

x Object of class [ibr](#).
... Further arguments passed to or from other methods.

Value

The function `plot.ibr` computes and returns a list of summary statistics of the fitted iterative bias reduction smoother given in `object`

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [summary.ibr](#)

Examples

```
## Not run: data(ozone, package = "ibr")
res.ibr <- ibr(ozone[,-1],ozone[,1],df=1.2)
plot(res.ibr)
## End(Not run)
```

poids

Product kernel evaluation

Description

Evaluate the product of kernel function at $(X\text{-valx})/bx$: Gaussian, Epanechnikov, Uniform, Quartic. This function is not intended to be used directly.

Usage

```
poids(kernelx,X,bx,valx,n,p)
```

Arguments

kernelx	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q").
X	Matrix of explanatory variables, size n, p .
bx	The vector of different bandwidths, length p .
valx	The vector of length p at which the product kernel is evaluated.
n	Number of rows of X .
p	Number of columns of X .

Value

Returns a vector which coordinates are the values of the product kernel at the given coordinate

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

See Also

[ibr](#)

`predict.ibr`*Predicted values using iterative bias reduction smoothers*

Description

Predicted values from iterative bias reduction object.
Missing values are not allowed.

Usage

```
## S3 method for class 'ibr'  
predict(object, newdata, interval=  
  c("none", "confidence", "prediction"), ...)
```

Arguments

<code>object</code>	Object of class <code>ibr</code> .
<code>newdata</code>	An optional matrix in which to look for variables with which to predict. If omitted, the fitted values are used.
<code>interval</code>	Type of interval calculation. Only <code>none</code> is currently available.
<code>...</code>	Further arguments passed to or from other methods.

Value

Produces a vector of predictions.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [summary.ibr](#)

Examples

```
## Not run: data(ozone, package = "ibr")  
res.ibr <- ibr(ozone[,-1],ozone[,1],df=1.2,K=1:500)  
summary(res.ibr)  
predict(res.ibr)  
## End(Not run)
```

predict.npregression *Predicted values using using local polynomials*

Description

Predicted values from a local polynomials of degree less than 2. See [locpoly](#) for fast binned implementation over an equally-spaced grid of local polynomial (gaussian kernel only)
Missing values are not allowed.

Usage

```
## S3 method for class 'npregression'  
predict(object, newdata, interval=  
  c("none", "confidence", "prediction"), deriv=FALSE, ...)
```

Arguments

object	Object of class npregression .
newdata	An optional vector of values to be predicted. If omitted, the fitted values are used.
interval	Type of interval calculation. Only none is currently available.
deriv	Boolean. If TRUE it returns the first derivative of the local polynomial (of degree 1).
...	Further arguments passed to or from other methods.

Value

Produces a vector of predictions. If `deriv` is TRUE the value is a named list with components: `yhat` which contains predictions and (if relevant) `deriv` the first derivative of the local polynomial of degree 1.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

See Also

[npregression](#), [summary.npregression](#), [locpoly](#)

Examples

```
f <- function(x){sin(5*pi*x)}
n <- 100
x <- runif(n)
z <- f(x)
sigma2 <- 0.05*var(z)
erreur<-rnorm(n,0,sqrt(sigma2))
y<-z+erreur
grid <- seq(min(x),max(x),length=500)
res <- npregression(x,y,bandwidth=0.02,control.par=list(degree=1))
plot(x,y)
lines(grid,predict(res,grid))
```

print.summary.ibr *Printing iterative bias reduction summaries*

Description

print method for class “summary.ibr”.

Usage

```
## S3 method for class 'summary.ibr'
print(x,displaybw=FALSE, digits =
max(3, getOption("digits") - 3), ...)
```

Arguments

x	Object of class <code>ibr</code> .
displaybw	Boolean that indicates if bandwidth are printed or not.
digits	Rounds the values in its first argument to the specified number of significant digits.
...	Further arguments passed to or from other methods.

Value

The function `print.summary.ibr` prints a list of summary statistics of the fitted iterative bias reduction model given in `x`.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [summary.ibr](#)

Examples

```
## Not run: data(ozone, package = "ibr")
res.ibr <- ibr(ozone[,-1],ozone[,1],df=1.2)
summary(res.ibr)
predict(res.ibr)
## End(Not run)
```

```
print.summary.npregression
```

Printing iterative bias reduction summaries

Description

print method for class "summary.npregression".

Usage

```
## S3 method for class 'summary.npregression'
print(x,digits =
max(3, getOption("digits") - 3), ...)
```

Arguments

x	Object of class npregression .
digits	Rounds the values in its first argument to the specified number of significant digits.
...	Further arguments passed to or from other methods.

Value

The function `print.summary.npregression` prints a list of summary statistics of the fitted iterative bias reduction model given in `x`.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

Examples

```
f <- function(x){sin(5*pi*x)}
n <- 100
x <- runif(n)
z <- f(x)
sigma2 <- 0.05*var(z)
erreur <- rnorm(n,0,sqrt(sigma2))
y <- z+erreur
res <- npregression(x,y,bandwidth=0.02)
summary(res)
```

summary.ibr

Summarizing iterative bias reduction fits

Description

summary method for class “ibr”.

Usage

```
## S3 method for class 'ibr'
summary(object, criteria="call", ...)
```

Arguments

object	Object of class <code>ibr</code> .
criteria	Character string which gives the criteria evaluated for the model. The criteria available are GCV (default, "gcv"), AIC ("aic"), corrected AIC ("aicc"), BIC ("bic") or gMDL ("gmdl"). The string "call" return the criterion used in the call of ibr.
...	Further arguments passed to or from other methods.

Value

The function `summary.ibr` computes and returns a list of summary statistics of the fitted iterative bias reduction smoother given in object

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#), [summary.ibr](#)

Examples

```
## Not run: data(ozone, package = "ibr")
res.ibr <- ibr(ozone[,-1],ozone[,1],df=1.2)
summary(res.ibr)
predict(res.ibr)
## End(Not run)
```

summary.npregression *Summarizing local polynomial fits*

Description

summary method for class “npregression”.

Usage

```
## S3 method for class 'npregression'
summary(object, criteria="call", ...)
```

Arguments

object	Object of class npregression .
criteria	Character string which gives the criteria evaluated for the model. The criteria available are GCV (default, "gcv"), AIC ("aic"), corrected AIC ("aicc"), BIC ("bic") or gMDL ("gmdl"). The string "call" return the criterion used in the call of npregression.
...	Further arguments passed to or from other methods.

Value

The function `summary.npregression` computes and returns a list of summary statistics of the local polynomial smoother given in `object`

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

See Also

[npregression](#), [summary.npregression](#)

Examples

```
f <- function(x){sin(5*pi*x)}
n <- 100
x <- runif(n)
z <- f(x)
sigma2 <- 0.05*var(z)
erreur <- rnorm(n,0,sqrt(sigma2))
y <- z+erreur
res <- npregression(x,y,bandwidth=0.02)
summary(res)
```

sumvalpr

Sum of a geometric series

Description

Calculates the sum of the first $(k+1)$ terms of a geometric series with initial term 1 and common ratio equal to valpr (lower or equal to 1).

Usage

```
sumvalpr(k,n,valpr,index1,index0)
```

Arguments

k	The number of terms minus 1.
n	The length of valpr.
valpr	Vector of common ratio in decreasing order.
index1	The index of the last common ratio equal to 1.
index0	The index of the first common ratio equal to 0.

Value

Returns the vector of the sums of the first $(k+1)$ terms of the geometric series.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

References

Cornillon, P. A., Hengartner, N. and Matzner-Lober, E. (2009) Recursive Bias Estimation for high dimensional regression smoothers. *submitted*.

See Also

[ibr](#)

tracekernel	<i>Trace of product kernel smoother</i>
-------------	---

Description

Evaluate the trace of the product of kernel smoother (Gaussian, Epanechnikov, Uniform, Quartic). This function is not intended to be used directly.

Usage

```
tracekernel(X,bx,kernelx,n,p)
```

Arguments

X	Matrix of explanatory variables, size n, p .
bx	The vector of different bandwidths, length p .
kernelx	Character string which allows to choose between gaussian kernel ("g"), Epanechnikov ("e"), uniform ("u"), quartic ("q").
n	Number of rows of X .
p	Number of columns of X .

Value

Evaluate the trace (effective degree of freedom) of the product kernel smoother.

Author(s)

Pierre-Andre Cornillon, Nicolas Hengartner and Eric Matzner-Lober.

See Also

[ibr](#)

Index

*Topic **datasets**

ozone, 38

*Topic **multivariate**

AIC.ibr, 3

betaA, 4

BIC, 6

calcA, 9

cvobs, 10

DuchonQ, 14

DuchonS, 15

fittedA, 15

forward, 17

ibr, 20

iterchoiceA, 23

iterchoiceAcv, 25

iterchoiceAcve, 26

iterchoiceAe, 28

iterchoiceS1, 29

iterchoiceS1cv, 30

iterchoiceS1cve, 31

iterchoiceS1e, 33

npregression, 36

plot.forwardibr, 39

plot.ibr, 40

predict.ibr, 42

predict.npregression, 43

print.summary.ibr, 44

print.summary.npregression, 45

summary.ibr, 46

summary.npregression, 47

sumvalpr, 48

*Topic **package**

ibr-package, 2

*Topic **smooth**

AIC.ibr, 3

betaA, 4

betaS1, 5

BIC, 6

bwchoice, 8

calcA, 9

cvobs, 10

departnoyau, 11

dssmoother, 12

dsSx, 13

DuchonQ, 14

DuchonS, 15

fittedA, 15

fittedS1, 16

forward, 17

ibr, 20

iterchoiceA, 23

iterchoiceAcv, 25

iterchoiceAcve, 26

iterchoiceAe, 28

iterchoiceS1, 29

iterchoiceS1cv, 30

iterchoiceS1cve, 31

iterchoiceS1e, 33

kernel, 34

kernelSx, 35

lambdchoice, 35

npregression, 36

plot.forwardibr, 39

plot.ibr, 40

poids, 41

predict.ibr, 42

predict.npregression, 43

print.summary.ibr, 44

print.summary.npregression, 45

summary.ibr, 46

summary.npregression, 47

sumvalpr, 48

tracekernel, 49

AIC.ibr, 3

AICc (BIC), 6

betaA, 4

betaS1, 5

- BIC, 6
- bwchoice, 8
- calca, 9
- critAaic (iterchoiceA), 23
- critAaicc (iterchoiceA), 23
- critAbic (iterchoiceA), 23
- critAgcv (iterchoiceA), 23
- critAgmdl (iterchoiceA), 23
- critS1aic (iterchoiceS1), 29
- critS1aicc (iterchoiceS1), 29
- critS1bic (iterchoiceS1), 29
- critS1gcv (iterchoiceS1), 29
- critS1gmdl (iterchoiceS1), 29
- cvobs, 10
- departnoyau, 11
- dssmoother, 12
- dsSx, 13
- DuchonQ, 14
- DuchonS, 15
- epane (kernel), 34
- fittedA, 15
- fittedS1, 16
- forward, 17, 40
- forwardibr, 39
- forwardibr (forward), 17
- gaussien (kernel), 34
- GCV (BIC), 6
- ibr, 3–11, 13–17, 20, 20, 25–27, 29–31, 33–42, 44–49
- ibr-package, 2
- image, 39
- iterchoiceA, 19, 22, 23, 25, 29
- iterchoiceAcv, 25
- iterchoiceAcve, 26
- iterchoiceAe, 22, 28
- iterchoiceS1, 19, 22, 29, 30, 34
- iterchoiceS1cv, 30
- iterchoiceS1cve, 31
- iterchoiceS1e, 22, 33
- kernel, 34
- kernelSx, 35
- lambdachoise, 35
- locpoly, 37, 38, 43
- npregression, 36, 43, 45, 47
- optimize, 19, 21, 24, 25, 29, 30
- ozone, 38
- plot.forwardibr, 20, 39
- plot.ibr, 40
- poids, 41
- predict.ibr, 23, 42
- predict.npregression, 38, 43
- print.ibr (ibr), 20
- print.npregression (npregression), 36
- print.summary.ibr, 44
- print.summary.npregression, 45
- quartic (kernel), 34
- residuals.ibr (ibr), 20
- residuals.npregression (npregression), 36
- scale, 22
- set.seed, 10, 25, 27, 30, 32
- summary.ibr, 4, 7, 23, 41, 42, 45, 46, 47
- summary.npregression, 38, 43, 47, 47
- sumvalpr, 48
- tracekernel, 49
- uniform (kernel), 34
- uniroot, 11, 18, 21