Package 'rmargint'

October 23, 2023

Type Package

Title Robust Marginal Integration Procedures

Version 2.0.3

Date 2023-10-23

Description Three robust marginal integration procedures for additive models based on local polynomial kernel smoothers. As a preliminary estimator of the multivariate function for the marginal integration procedure, a first approach uses local constant M-estimators, a second one uses local polynomials of order 1 over all the components of covariates, and the third one uses M-estimators based on local polynomials but only in the direction of interest. For this last approach, estimators of the derivatives of the additive functions can be obtained. All three procedures can compute predictions for points outside the training set if desired. See Boente and Martinez (2017) <doi:10.1007/s11749-016-0508-0> for details.

License GPL (>= 3.0)

RoxygenNote 7.2.3

Encoding UTF-8

Imports stats, graphics

NeedsCompilation yes

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

Maintainer Alejandra Martinez <ale_m_martinez@hotmail.com>

Repository CRAN

Date/Publication 2023-10-23 21:00:02 UTC

R topics documented:

`rmargint-package` *Robust marginal integration estimators for additive models.*

Description

Robust marginal integration estimators for additive models.

Author(s)

Alejandra Martinez, Matias Salibian-Barrera

Maintainer: Alejandra Martinez <ale_m_martinez@hotmail.com>

References

Boente G. and Martinez A. (2017). Marginal integration M-estimators for additive models. TEST, 26, 231-260.

`deviance.margint` *Deviance for objects of class* `margint`

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 `margint.cl` or `margint.rob`.

Usage

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

Arguments

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

Value

A real number.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

---

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

---

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 margint.cl or margint.rob.

Usage

```
## S3 method for class 'margint'
fitted.values(object, ...)
```

Arguments

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

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

formula.margint *Additive model formula*

Description

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

Usage

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

Arguments

`x`	an object of class `margint`, a result of a call to `margint.cl` or `margint.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)
```

---

kernel10 *Order 10 kernel*

Description

This function evaluates a kernel of order 10. A kernel of order 10.

Usage

```
kernel10(x)
```

Arguments

x	A vector of real numbers.

Details

This function evaluates a kernel of order 10. A kernel L is a kernel of order 10 if it integrates 1, the integrals of u^j L(u) are 0 for 1 <= j < 10 (j integer) and the integral of u^10 L(u) is different from 0.

Value

A vector of the same length as x where each entry is `0.75 * ( 1 - x^2 ) * ( 315/128 - 105/32 * x^2 + 63/64 * x^4 - 3/32 * x^6 - 1/384 * x^8 )` and 0 otherwise.

Author(s)

Alejandra Martinez, <ale_m_martinez@hotmail.com>, Matias Salibian-Barrera

Examples

```
x <- seq(-2,2,length=10)
kernel10(x)
```

kernel4 *Order 4 kernel*

Description

This function evaluates a kernel of order 4.

Usage

```
kernel4(x)
```

Arguments

x	A vector of real numbers.

Details

This function evaluates a kernel of order 4. A kernel L is a kernel of order 4 if it integrates 1, the integrals of u^j L(u) are 0 for 1 <= j < 4 (j integer) and the integral of u^4 L(u) is different from 0.

Value

A vector of the same length as `x` where each entry is `( 15/32 ) * ( 1 - x^2 ) * ( 3 - 7 * x^2 )` if `abs(x) < 1` and 0 otherwise.

Author(s)

Alejandra Martinez, <ale_m_martinez@hotmail.com>, Matias Salibian-Barrera

Examples

```
x <- seq(-2,2,length=10)
kernel4(x)
```

kernel6 *Order 6 kernel*

Description

This function evaluates a kernel of order 6.

Usage

```
kernel6(x)
```

Arguments

x	A vector of real numbers.

Details

This function evaluates a kernel of order 6. A kernel L is a kernel of order 6 if it integrates 1, the integrals of u^j L(u) are 0 for 1 <= j < 6 (j integer) and the integral of u^6 L(u) is different from 0.

Value

A vector of the same length as x where each entry is `( 105/256 ) * ( 1 - x^2 ) * ( 5 - 30 * x^2 + 33 * x^4 )` if `abs(x) < 1` and 0 otherwise.

Author(s)

Alejandra Martinez, <ale_m_martinez@hotmail.com>, Matias Salibian-Barrera

Examples

```
x <- seq(-2,2,length=10)
kernel6(x)
```

---

kernel8 *Order 8 kernel*

Description

This function evaluates a kernel of order 8.

Usage

```
kernel8(x)
```

Arguments

x	A vector of real numbers.

Details

This function evaluates a kernel of order 8. A kernel L is a kernel of order 8 if it integrates 1, the integrals of u^j L(u) are 0 for 1 <= j < 8 (j integer) and the integral of u^8 L(u) is different from 0.

Value

A vector of the same length as x where each entry is `( 315/4096 ) * ( 1 - x^2 ) * ( 35 - 385 * x^2 + 1001 * x^4 - 715 * x^6 )` and 0 otherwise.

Author(s)

Alejandra Martinez, <ale_m_martinez@hotmail.com>, Matias Salibian-Barrera

Examples

```
x <- seq(-2,2,length=10)
kernel8(x)
```

`margint.cl` *Classic marginal integration procedures for additive models*

Description

This function computes the standard marginal integration procedures for additive models.

Usage

```
margint.cl(
  formula,
  data,
  subset,
  point = NULL,
  windows,
  epsilon = 1e-06,
  prob = NULL,
  type = "0",
  degree = NULL,
  qderivate = FALSE,
  orderkernel = 2,
  Qmeasure = NULL
)
```

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`	a matrix of points where predictions will be computed and returned.
`windows`	a vector or a squared matrix of bandwidths for the smoothing estimation procedure.

epsilon	convergence criterion.
prob	a vector of probabilities of observing each response (n). Defaults to NULL.
type	three different type of estimators can be selected: type '0' (local constant on all the covariates), type '1' (local linear smoother on all the covariates), type 'alpha' (local polynomial smoother only on the direction of interest).
degree	degree of the local polynomial smoother in the direction of interest when using the estimator of type 'alpha'. Defaults to NULL for the case when using estimators of type '0' or '1'.
qderivate	if TRUE, it calculates g^(q+1)/(q+1)! for each component only for the type 'alpha' method. Defaults to FALSE.
orderkernel	order of the kernel used in the nuisance directions when using the estimator of type 'alpha'. Defaults to 2.
Qmeasure	a matrix of points where the integration procedure ocurrs. Defaults to NULL for calcuting the integrals over the sample.

Details

This function computes three types of classical marginal integration procedures for additive models, that is, considering a squared loss function.

Value

A list with the following components:

mu	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.
mul	A vector of size p showing in each component the estimated intercept that considers only that direction of interest when using the type 'alpha' method.
g.derivative	Matrix of estimated derivatives of the additive components (only when qderivate is TRUE) (n by p).
prediction.derivate	Matrix of estimated derivatives of the additive components for the points listed in the argument point (only when qderivate is TRUE).
Xp	Matrix of explanatory variables.
yp	Vector of responses.
formula	Model formula

Author(s)

Alejandra Martinez, <ale_m_martinez@hotmail.com>, Matias Salibian-Barrera

References

Chen R., Hardle W., Linton O.B. and Severance-Lossin E. (1996). Nonparametric estimation of additive separable regression models. Physica-Verlag HD, Switzerland. Linton O. and Nielsen J. (1995). A kernel method of estimating structured nonparametric regression based on marginal integration. Biometrika, 82(1), 93-101. Severance-Lossin E. and Sperlich S. (1999). Estimation of derivatives for additive separable models. Statistics, 33(3), 241-265. Tjostheim D. and Auestad B. (1994). Nonparametric identification of nonlinear time series: Selecting significant lags. Journal of the American Statistical Association, 89(428), 1410-1430.

Examples

```
function.g1 <- function(x1) 24*(x1-1/2)^2-2
function.g2 <- function(x2) 2*pi*sin(pi*x2)-4
n <- 150
x1 <- runif(n)
x2 <- runif(n)
X <- cbind(x1, x2)
eps <- rnorm(n,0,sd=0.15)
regresion <- function.g1(x1) + function.g2(x2)
y <- regresion + eps
bandw <- matrix(0.25,2,2)
set.seed(8090)
nQ <- 80
Qmeasure <- matrix(runif(nQ*2), nQ, 2)
fit.cl <- margint.cl(y ~ X, windows=bandw, type='alpha', degree=1, Qmeasure=Qmeasure)
```

`margint.rob` *Robust marginal integration procedures for additive models*

Description

This function computes robust marginal integration procedures for additive models.

Usage

```
margint.rob(
  formula,
  data,
  subset,
  point = NULL,
  windows,
  prob = NULL,
  sigma.hat = NULL,
  win.sigma = NULL,
  epsilon = 1e-06,
  type = "0",
  degree = NULL,
```

```
  typePhi = "Huber",
  k.h = 1.345,
  k.t = 4.685,
  max.it = 20,
  qderivate = FALSE,
  orderkernel = 2,
  Qmeasure = NULL
)
```

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	a matrix of points where predictions will be computed and returned.
windows	a vector or a squared matrix of bandwidths for the smoothing estimation procedure.
prob	a vector of probabilities of observing each response (n). Defaults to NULL.
sigma.hat	estimate of the residual standard error. If NULL we use the mad of the residuals obtained with local medians.
win.sigma	a vector of bandwidths for estimating sigma.hat. If NULL it uses the argument windows if it is a vector or its diagonal if it is a matrix.
epsilon	convergence criterion.
type	three different type of estimators can be selected: type '0' (local constant on all the covariates), type '1' (local linear smoother on all the covariates), type 'alpha' (local polynomial smoother only on the direction of interest).
degree	degree of the local polynomial smoother in the direction of interest when using the estimator of type 'alpha'. Defaults to NULL for the case when using estimators of type '0' or '1'.
typePhi	one of either 'Tukey' or 'Huber'.
k.h	tuning constant for a Huber-type loss function. Defaults to 1.345.
k.t	tuning constant for a Tukey-type loss function. Defaults to 4.685.
max.it	maximum number of iterations for the algorithm.
qderivate	if TRUE, it calculates g^(q+1)/(q+1)! for each component only for the type 'alpha' method. Defaults to FALSE.
orderkernel	order of the kernel used in the nuisance directions when using the estimator of type 'alpha'. Defaults to 2.
Qmeasure	a matrix of points where the integration procedure ocurrs. Defaults to NULL for calcuting the integrals over the sample.

Details

This function computes three types of robust marginal integration procedures for additive models.

Value

A list with the following components:

`mu`	Estimate for the intercept.
`g.matrix`	Matrix of estimated additive components (n by p).
`sigma.hat`	Estimate of the residual standard error.
`prediction`	Matrix of estimated additive components for the points listed in the argument point.
`mul`	A vector of size p showing in each component the estimated intercept that considers only that direction of interest when using the type `'alpha'` method.
`g.derivative`	Matrix of estimated derivatives of the additive components (only when qderivate is `TRUE`) (n by p).
`prediction.derivate`	Matrix of estimated derivatives of the additive components for the points listed in the argument point (only when qderivate is `TRUE`).
`Xp`	Matrix of explanatory variables.
`yp`	Vector of responses.
`formula`	Model formula

Author(s)

Alejandra Martinez, <ale_m_martinez@hotmail.com>, Matias Salibian-Barrera

References

Boente G. and Martinez A. (2017). Marginal integration M-estimators for additive models. TEST, 26(2), 231-260. https://doi.org/10.1007/s11749-016-0508-0

Examples

```
function.g1 <- function(x1) 24*(x1-1/2)^2-2
function.g2 <- function(x2) 2*pi*sin(pi*x2)-4
set.seed(140)
n <- 150
x1 <- runif(n)
x2 <- runif(n)
X <- cbind(x1, x2)
eps <- rnorm(n,0,sd=0.15)
regresion <- function.g1(x1) + function.g2(x2)
y <- regresion + eps
bandw <- matrix(0.25,2,2)
set.seed(8090)
nQ <- 80
Qmeasure <- matrix(runif(nQ*2), nQ, 2)
```

```
fit.rob <- margint.rob(y ~ X, windows=bandw, type='alpha', degree=1, Qmeasure=Qmeasure)
```

`my.norm.2` *Euclidean norm of a vector*

Description

This function calculates the Euclidean norm of a vector.

Usage

```
my.norm.2(x)
```

Arguments

`x`	A real vector.

Value

The Euclidean norm of the input vector.

Author(s)

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

Examples

```
x <- seq(-2, 2, length=10)
my.norm.2(x)
```

`plot.margint` *Diagnostic plots for objects of class* `margint`

Description

Plot method for class `margint`.

Usage

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

Arguments

x	an object of class margint, a result of a call to margint.cl or margint.rob.
derivative	if TRUE, it plots the q-th derivatives. Defaults to FALSE.
which	vector of indices of explanatory variables for which partial residuals plots will be generated. Defaults to all available explanatory variables.
ask	logical value. If TRUE, the graphical device will prompt before going to the next page/screen of output.
...	additional other arguments.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

Examples

```
function.g1 <- function(x1) 24*(x1-1/2)^2-2
function.g2 <- function(x2) 2*pi*sin(pi*x2)-4
set.seed(140)
n <- 150
x1 <- runif(n)
x2 <- runif(n)
X <- cbind(x1, x2)
eps <- rnorm(n,0,sd=0.15)
regresion <- function.g1(x1) + function.g2(x2)
y <- regresion + eps
bandw <- matrix(0.25,2,2)
set.seed(8090)
nQ <- 80
Qmeasure <- matrix(runif(nQ*2), nQ, 2)
fit.rob <- margint.rob(y ~ X, windows=bandw, type='alpha', degree=1, Qmeasure=Qmeasure)
plot(fit.rob, which=1)
```

---

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

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 margint.cl or margint.rob.

Usage

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

Arguments

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

Value

A vector of fitted values.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

---

print.margint *Print a Marginal Integration procedure*

Description

The default print method for a margint object.

Usage

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

Arguments

x	an object of class margint, a result of a call to margint.cl or margint.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 r.

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 r.

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.margint *Residuals for objects of class* margint

Description

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

Usage

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

Arguments

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

Value

A vector of residuals.

Author(s)

Alejandra Mercedes Martinez <ale_m_martinez@hotmail.com>

summary.margint *Summary for additive models fits using a marginal integration procedure*

Description

Summary method for class margint.

Usage

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

Arguments

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

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