Package 'eBsc'

January 11, 2021

Type Package

Title ``Empirical Bayes Smoothing Splines with Correlated Errors''

Version 4.13

Date 2021-01-06

Author Francisco Rosales, Tatyana Krivobokova, Paulo Serra.

Description Presents a statistical method that uses a recursive algorithm for signal extraction. The method handles a non-parametric estimation for the correlation of the errors. See ``Serra", ``Krivobokova" and ``Rosales" (2018) <arXiv:1812.06948> for details.

License GPL-2

Imports Brobdingnag, parallel, nlme, Matrix, MASS, splines, Rcpp

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Maintainer John Barrera <johnkevinbarrera@gmail.com>

Repository CRAN

Date/Publication 2021-01-11 06:50:09 UTC

R topics documented:

eBsc-package *Empirical Bayes Smoothing Splines with Correlated Errors*

Description

Empirical Bayes smoothing splines with correlated errors. The method uses a recursive algorithm for signal extraction with a non-parametric estimation of the correlation matrix of the errors.

Details

Package:	eBsc
Version:	4.11
Date:	2020-08-21
Depends:	Brobdingnag, parallel, nlme, Matrix, MASS

Index:

`eBsc`	`Empirical Bayes smoothing splines with correlated errors`
`plot.eBsc`	`Plots fitted curves from the filter`
`summary.eBsc`	`Summary information of the error`

The function `eBsc()` is used to fit the model. Using the resulting `eBsc` object and summary information on the errors can be printed using `summary`.

Author(s)

Francisco Rosales, Paulo Serra, Tatyana Krivobokova Maintainer: Francisco Rosales <francisco.rosales-marticorena@protonmail.com>

References

Serra, P. and Krivobokova, T. (2015)
Adaptive Empirical Bayesian Smoothing Splines

See Also

`stl` (package stats), `HoltWinters` (package stats)

Examples

```
# simulated data for non-correlated errors
library(eBsc)
n <- 250
sigma <- 0.05
beta <- function(x,p,q){
gamma(p+q)/(gamma(p)*gamma(q))*x^(p-1)*(1-x)^(q-1)
```

```
}
x <- seq(0, 1, length.out = n)
mu <- (6 * beta(x, 30, 17) + 4 * beta(x, 3, 11))/10;
mu <- (mu - min(mu))/(max(mu) - min(mu))
noise <- rnorm(n)
y <- mu + sigma * noise

# correlation matrix assumed known and equal to the identity
fit.d <- eBsc(y, method = "D", R0 = diag(n))
plot(fit.d, full = FALSE)
```

---

drbasis *Computation of the Demmler-Reinsch basis.*

Description

The Demmler-Reinsch basis is provided for a given smoothness degree in a uniform grid.

Usage

```
drbasis(nn,qq)
```

Arguments

nn	Number of design points in the uniform grid.
qq	Smoothness degree of the basis.

Details

The use of large numbers required by the basis is handled by the package Brobdingnag. The method assumes the grid is equidistant. Missing values are not supported.

Value

A list object containing the following information.

eigenvalues	estimadted eigenvalues
eigenvectors	estimated eigenvectors
eigenvectorsQR	orthonormal eigenvectors
x	equidistant grid used to build the basis

Author(s)

Francisco Rosales

References

Rosales F. (2016).
Empirical Bayesian Smoothing Splines for Signals with Correlated Errors: Methods and Applications

Serra, P. and Krivobokova, T. (2015)
Adaptive Empirical Bayesian Smoothing Splines

Examples

```
oldpar <- par(no.readonly = TRUE)
#plot elements of the basis
library(eBsc)
n <- 100
Basis <- list()
for(i in 1:6){Basis[[i]] <- drbasis( nn = n, qq = i)}

#eigenvalues
par(mfrow = c(3,2), mar = c(4,2,2,2))
for(i in 1:6){
name <- paste("Eigenvalues (q = ",i,")", sep = "")
plot(Basis[[i]]$eigenvalues,
type = 'l', lwd = 2, xlab = "x", ylab = "", main = name)
}
par(oldpar)

#eigenvectors for q = 3
par(mfrow = c(3,2), mar = c(4,2,2,2))
for(i in 1:6){
name <- paste("Eigenvector n. ", i + 3, sep = "")
plot(Basis[[i]]$eigenvectorsQR[, i + 3],
type = 'l', lwd = 2, xlab = "x", ylab = "", main = name)
}
par(oldpar)

#example of a smooth function in the Demmler-Reinsch basis
library(eBsc)
n <- 200
Basis <- list()
for(i in 1:6){Basis[[i]] <- drbasis(nn = n, qq = i)}
coef3 <- c(rep(0,3), (pi*(2:(n-2))) ^ (-3.1)) * (cos(2*(1:n)))
A3 <- Basis[[3]]$eigenvectors
mu <- -A3%*%coef3
mu <- (mu - min(mu)) / (max(mu) - min(mu))
plot(mu, xlab = "x", ylab = "mu", type = 'l', lwd = 2)
par(oldpar)
```

Description

Empirical Bayes smoothing splines with correlated errors. The method uses a recursive algorithm for signal extraction with a non-parametric estimation of the correlation matrix of the errors.

Usage

```
eBsc(y, q, method, parallel, R0, zero_range, ARpMAq, trace, tol.lambda, tol.rho, max.iter)
```

Arguments

y	Is a univariate numeric vector without missing values.
q	Is the value of q if known. If left empty the method considers all possibles q's between 1 and 6 and selects the best one according to the Tq criteria. q=NULL is the default.
method	Is a method used for the fit. It can take the values "D" (deterministic fit), "P" (parametric fit) and "N" (non-parametric fit). For example: i) to fit a model with known correlation matrix R.known one should select method = "D" and R0 = R.known; ii) to fit a model with a nonparametric estimation of the correlation and a starting correlation matrix R.start, one should select method = "N" and R0 = R.start; and iii) to fit a model with an ARMA parametric structure R.ARMA, one should select method="P" and ARpMAq=c(1,0). method = "N" is the default.
parallel	Is a logical parameter indicating if parallel computation should be used. parallel=FALSE is the default.
R0	Is the starting correlation matrix. If method = "D" this matrix is not changed by the algorithm.
zero_range	Is the interval to look for zeros in the estimating equation for the smoothing parameter (lambda).
ARpMAq	Is the desired ARMA structure for the noise process.
trace	If true, the process of the algorithm is traced and reported.
tol.lambda	Tolerance level for lambda.
tol.rho	Tolerance level for rho.
max.iter	Maximum number of iterations.

Details

The method assumes the data is equidistant.

Value

A list object of class eBsc containing the following information.

q.hat	estimadted q
lambda.hat	estimated lambda
R.hat	estimated correlation matrix

f.hat	estimated function
f.hat	estimated variance
etq.hat	estimating equation for q
data	data used to fit the model
call	Call of eBsc

Author(s)

Francisco Rosales, Paulo Serra, Tatyana Krivobokova

References

Serra, P. and Krivobokova, T. (2015)
Adaptive Empirical Bayesian Smoothing Splines

See Also

stl (package stats), HoltWinters (package stats)

Examples

```
library(eBsc)
n <- 250
sigma <- 0.05
beta <- function(x,p,q){
  gamma(p+q)/(gamma(p)*gamma(q))*x^(p-1)*(1-x)^(q-1)
}
x <- seq(0, 1, length.out = n)
mu <- (6 * beta(x, 30, 17) + 4 * beta(x, 3, 11))/10;
mu <- (mu - min(mu))/(max(mu) - min(mu))
noise <- rnorm(n)
y <- mu + sigma * noise

# correlation matrix assumed known and equal to the identity
fit <- eBsc(y,  method = "D", R0 = diag(n))
plot(fit, full=FALSE)
```

---

plot.eBsc *Plot fitted components*

Description

Plot fitted components and the acf of the errors.

Usage

```
## S3 method for class 'eBsc'
plot(x,full=FALSE,...)
```

Arguments

x	eBsc object.
full	plot option. If TRUE graphial details of the estimation are provided. If FALSE a simple plot of the estimation and its confidence bands is provided.
...	further arguments to be passed to plot().

Details

if the eBsc plots the fits and the acf of the errors.

Value

The function returns the selected plots.

Author(s)

Francisco Rosales, Paulo Serra, Tatyana Krivobokova.

References

Serra, P. and Krivobokova, T. (2015)
Adaptive Empirical Bayesian Smoothing Splines

Examples

```
library(eBsc)
n <- 250
sigma <- 0.05
Basis <- list()
for(i in 1:6) Basis[[i]] <- drbasis(nn = n, qq = i)
coef3 <- c(rep(0,3),(pi*(2:(n-2)))^(-3.1))*(cos(2*(1:n)))
A3 <- Basis[[3]]$eigenvectors
mu <- A3%*%coef3
mu <- (mu-min(mu))/(max(mu)-min(mu))
noise <- rnorm(n)
y <- mu + sigma * noise

#correlation assumed known and equal to the identity
fit.d <- eBsc(y, method = "D", R0 = diag(n))

#simple plot by
plot(fit.d, full = FALSE)
```

RcppArmadillo-Functions

Set of functions in example RcppArmadillo package

Description

These four functions are created when `RcppArmadillo.package.skeleton()` is invoked to create a skeleton packages.

Usage

```
rcpparma_hello_world()
rcpparma_outerproduct(x)
rcpparma_innerproduct(x)
rcpparma_bothproducts(x)
```

Arguments

`x`	a numeric vector

Details

These are example functions which should be largely self-explanatory. Their main benefit is to demonstrate how to write a function using the Armadillo C++ classes, and to have to such a function accessible from R.

Value

`rcpparma_hello_world()` does not return a value, but displays a message to the console.

`rcpparma_outerproduct()` returns a numeric matrix computed as the outer (vector) product of `x`.

`rcpparma_innerproduct()` returns a double computer as the inner (vector) product of `x`.

`rcpparma_bothproducts()` returns a list with both the outer and inner products.

Author(s)

Dirk Eddelbuettel

References

See the documentation for Armadillo, and RcppArmadillo, for more details.

Examples

```
x <- sqrt(1:4)
rcpparma_innerproduct(x)
rcpparma_outerproduct(x)
```

summary.eBsc *eBsc Summary*

Description

Takes an eBsc object produced by eBsc and summarizes the information of the errors.

Usage

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

Arguments

object	eBsc object.
...	further arguments to be passed to summary().

Value

The function gives basic statistics of the error from applying eBsc.

Author(s)

Francisco Rosales, Paulo Serra, Tatyana Krivobokova

References

Serra, P. and Krivobokova, T. (2015)
Adaptive Empirical Bayesian Smoothing Splines

See Also

plot.eBsc (package eBsc),

Examples

```
# simulated data
library(eBsc)
n <- 250
sigma <- 0.05

Basis <- list()
for(i in 1:6){Basis[[i]] <- drbasis(nn = n, qq = i)}
coef3 <- c(rep(0,3),(pi*(2:(n-2)))^(-3.1)) * (cos(2*(1:n)))
A3 <- Basis[[3]]$eigenvectors
mu <- - A3%*%coef3
mu <- (mu - min(mu))/(max(mu) - min(mu))
noise <- rnorm(n)
y <- mu + sigma * noise
```

```
# correlation matrix assumed known and equal to the identity
fit <- eBsc(y, method = "D", R0 = diag(n))

summary(fit)
```

Index