Package ‘eBsc’

September 1, 2020

Type    Package
Title   Empirical Bayes Smoothing Splines with Correlated Errors
Version 4.11
Date    2020-08-21
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. For details on the methods see: Serra, Krivobokova and Rosales (2018) <arXiv:1812.06948>.
License GPL-2
Imports Brobdingnag, parallel, nlme, Matrix, MASS, splines, Rcpp
LinkingTo Rcpp, RcppArmadillo
NeedsCompilation yes
Author  Francisco Rosales [aut, cre],
        Tatyana Krivobokova [aut],
        Paulo Serra [aut]
Maintainer Francisco Rosales <francisco@brein.pe>
Repository CRAN
Date/Publication 2020-09-01 08:40:13 UTC

R topics documented:

eBsc-package ......................................................... 2
drbasis ............................................................... 3
eBsc ................................................................. 4
plot.eBsc ........................................................... 6
RcppArmadillo-Functions .......................................... 8
summary.eBsc ...................................................... 9

Index 11
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

Adaptive Empirical Bayesian Smoothing Splines

See Also

- stl (package stats), HoltWinters (package stats)

Examples

```r
# 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)
```

```
\begin{verbatim}

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

# correlation matrix assumed known and equal to the identity
fit.d <- ebse(y, method = "D", R0 = diag(n))
plot(fit.d, full = FALSE)
\end{verbatim}

---

drbasis

\textit{Computation of the Demmler-Reinsch basis.}

\section*{Description}

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

\section*{Usage}

\texttt{drbasis(nn, qq)}

\section*{Arguments}

\begin{itemize}
  \item \texttt{nn}  
  \hspace{1em} Number of design points in the uniform grid.
  \item \texttt{qq}  
  \hspace{1em} Smoothness degree of the basis.
\end{itemize}

\section*{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.

\section*{Value}

A list object containing the following information.

\begin{itemize}
  \item \texttt{eigenvalues}  
  \hspace{1em} estimated eigenvalues
  \item \texttt{eigenvectors}  
  \hspace{1em} estimated eigenvectors
  \item \texttt{eigenvectorsQR}  
  \hspace{1em} orthonormal eigenvectors
  \item \texttt{x}  
  \hspace{1em} equidistant grid used to build the basis
\end{itemize}

\section*{Author(s)}

Francisco Rosales
References

Empirical Bayesian Smoothing Splines for Signals with Correlated Errors: Methods and Applications

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 estimated q
lambda.hat estimated lambda
R.hat estimated correlation matrix
plot.eBsc

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
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.
plot.eBsc

Usage

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

Arguments

x      eBsc object.
full   plot option. If TRUE graphical 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

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)
Description

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

Usage

```r
## 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

Adaptive Empirical Bayesian Smoothing Splines

See Also

`plot.eBsc` (package eBsc),

Examples

```r
# 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

* Demmler-Reinsch basis
drbasis, 3
* bayesian
eBsc, 4
* correlated errors
eBsc, 4
  summary.eBsc, 9
* non-parametric
drbasis, 3
eBsc, 4
* package
eBsc-package, 2
* plot
  plot.eBsc, 6
* summary
  summary.eBsc, 9
drbasis, 3
eBsc, 4
eBsc-package, 2
HoltWinters, 2, 6
plot.eBsc, 6, 9
rcpparma_bothproducts
  RcppArmadillo-Functions, 8
rcpparma_hello_world
  RcppArmadillo-Functions, 8
rcpparma_innerproduct
  RcppArmadillo-Functions, 8
rcpparma_outerproduct
  RcppArmadillo-Functions, 8
RcppArmadillo-Functions, 8
stl, 2, 6
summary, 2
summary.eBsc, 9