Package ‘MICsplines’

September 7, 2021

Type Package

Version 1.0

Date 2021-08-25

Title The Computing of Monotonic Spline Bases and Constrained Least-Squares Estimates

Author Yili Hong [aut, cre], Jie Min [aut, ctb]

Maintainer Yili Hong <yilihong@vt.edu>

Description Providing C implementation for the computing of monotonic spline bases, including M-splines, I-splines, and C-splines, denoted by MIC splines. The definitions of the spline bases are described in Meyer (2008) <doi:10.1214/08-AOAS167>. The package also provides the computing of constrained least-squares estimates when a subset of or all of the regression coefficients are constrained to be non-negative.

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-09-07 13:30:05 UTC

R topics documented:

MICsplines-package 2
clse 2
MIC.splines.basis.fast 3

Index **5**

MICsplines-package *The Computing of Monotonic Spline Bases and Constrained Least-Squares Estimates*

Description

The package provides C implementation for the computing of monotonic spline bases, including M-splines, I-splines, and C-splines, denoted by MIC splines. The definitions of the spline bases are described in Meyer (2008). The package also provides the computing of constrained least-squares estimates when a subset of or all of the regression coefficients are constrained to be non-negative, as described in Fraser and Massam (1989).

References

Fraser, D. A. S. and H. Massam (1989). A mixed primal-dual bases algorithm for regression under inequality constraints. Application to concave regression. *Scandinavian Journal of Statistics* 16, 65-74.

Meyer, M. C. (2008). Inference using shape-restricted regression splines. *The Annals of Applied Statistics* 2, 1013-1033.

clse *Constrained Least-Squares Estimates*

Description

This function computes the constrained least-squares estimates when a subset of or all of the regression coefficients are constrained to be non-negative, as described in Fraser and Massam (1989).

Usage

```
clse(dat.obj)
```

Arguments

`dat.obj` A list with the following format, `list(y,mat,lam)`. Here `y` is the response vector, `mat` is the design matrix for the regression, and `lam` is a vector with the length that matches the number of columns in `mat`. The values of `lam` is either 0 or 1, with 0 means unconstrained and 1 means the corresponding regression coefficient is constrained to be non-negative.

Value

The returned value is a list with format, `list(dat.obj,beta.vec,yhat)`. Here `dat.obj` is the input of the function, `beta.vec` gives the estimated regression coefficient, and `yhat` is the vector for the fitted response values.

References

Fraser, D. A. S. and H. Massam (1989). A mixed primal-dual bases algorithm for regression under inequality constraints. Application to concave regression. *Scandinavian Journal of Statistics* 16, 65-74.

Examples

```
#generate a dataset for illustration.
x=seq(1,10,,100)
y=x^2+rnorm(length(x))
#generate spline bases.
tmp=MIC.splines.basis.fast(x=x, df = 10, knots = NULL, boundary.knots=NULL,
type="Is",degree = 3,delta=0.001,eq.alloc=FALSE)
#plot the spline bases.
plot(tmp)
#generate the data object for the clse function.
dat.obj=list(y=y, mat=cbind(1, tmp$mat), lam=c(0, rep(1, ncol(tmp$mat))))
#fit clse.
fit=clse(dat.obj=dat.obj)
#visualize fitted results.
plot(x, y, pch=16)
lines(x, fit$yhat, lwd=3, col=2)
```

MIC.splines.basis.fast

Generating MIC Spline Bases

Description

This function provides C implementation for the computing of monotonic spline bases, including M-splines, I-splines, and C-splines, denoted by MIC splines. The definitions of the spline bases are described in Meyer (2008).

Usage

```
MIC.splines.basis.fast(x, df = NULL, knots = NULL, boundary.knots = NULL,
type = "Ms", degree = 3, delta = 0.01, eq.alloc = FALSE)
```

Arguments

`x`	A numeric vector for the data to generate spline bases for.
`df`	The degree of freedom, which equals to the number of interior knots plus the spline degree.
`knots`	A vector for the interior knots.
`boundary.knots`	The values for the left and right boundary points.

`type`	The type of splines to be computed. `"Ms"` stands for M-splines, `"Is"` stands for I-splines, `"IsN"` stands for I-splines without normalization, and `"Cs"` stands for C-splines.
`degree`	The degree for the M-splines. I-splines are based on the integration of the M-splines, and C-splines are based on the integration of the I-splines.
`delta`	A numeric value that is used to set the bin width for numerical integration. Usually it is set to a small number.
`eq.alloc`	A logic variable, which is true if using equal spacing for the interior knots, and is false if using equal quantiles for the interior knots.

Value

A list with format, `list(mat,x,...)`. Here `mat` is the matrix for the spline bases, `x` is the vector for the data, and the rest of the items are carrying the information from the arguments.

References

Meyer, M. C. (2008). Inference using shape-restricted regression splines. *The Annals of Applied Statistics* 2, 1013-1033.

Examples

```
#generate a dataset for illustration.
x=seq(1,10,,100)
y=x^2+rnorm(length(x))
#generate spline bases.
tmp=MIC.splines.basis.fast(x=x, df = 10, knots = NULL, boundary.knots=NULL,
type="Is",degree = 3,delta=0.001,eq.alloc=FALSE)
#plot the spline bases.
plot(tmp)
```

Index

∗ **package**
    MICsplines-package, 2

clse, 2

MIC.splines.basis.fast, 3
MICsplines-package, 2