Package 'orthogonalsplinebasis'

May 23, 2022

Type Package

Title Orthogonal B-Spline Basis Functions

Version 0.1.7

Date 2022-05-10

Author Andrew Redd

Maintainer Andrew Redd <andrew.redd@hsc.utah.edu>

Description Represents the basis functions for B-splines in a simple matrix formulation that facilitates, taking integrals, derivatives, and making orthogonal the basis functions.

License GPL (>= 2)

URL https://github.com/halpo/obsplines

Suggests spelling

Language en-US

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-23 19:00:01 UTC

R topics documented:

`orthogonalsplinebasis-package` *A Matrix Representation for Spline Basis Functions*

Description

This package provides functions for manipulation of spline basis functions. A matrix representation for the basis functions is at the center of the functions. The matrix representation simplifies the process of orthogonalization as well as differentiation and integration.

Author(s)

Andrew Redd Maintainer: Andrew Redd <andrew.redd@hsc.utah.edu>

`evaluate-methods` *Generic evaluate method*

Description

Methods for function `evaluate`.

Methods

object = "SplineBasis", x = "numeric" Evaluates a `SplineBasis` object for the spline basis curves at points `x`. See SplineBasis

---

`expand.knots` *Expands knots for appropriate number of knots in B-splines*

Description

This function is for convenience of specifying knots for B-splines. Since the user usually only want to specify the interval that they are interested in the end knots are usually duplicated. This function interprets the first and last knots as the end points and duplicates them.

Usage

```
expand.knots(interior, order = 4)
```

Arguments

interior	The knots including all interior and endpoint knots
order	the order of the splines that the knots are to be used with. Defaults to 4, being cubic splines

Value

A vector of knots with the order specified as an attribute

Author(s)

Andrew Redd

See Also

SplineBasis, ~~~

Examples

```
(knots<-expand.knots(1:10))
plot(OBasis(knots))
```

---

fitLS *Fitting splines with penalized least squares.*

Description

Estimates the control vector for a spline fit by penalized least squares. The penalty being the penalty parameter times the functional inner product of the second derivative of the spline curve.

Usage

```
fitLS(object, x, y, penalty = 0)
```

Arguments

object	The SplineBasis object to be used to make the fit
x	predictor variable.
y	response variable.
penalty	The penalty multiplier.

Details

For numeric vector y, and x, and a set of basis functions, represented in `object`, defined on the knots $(k_0, \ldots, k_m)$. The likelihood is defined by

$$\sum_{i=1}^{n}(y_i - b(x_i)\mu) + \int_{k_0}^{k_m} \mu^T b''(t)^T b''(t)\mu dt$$

The function estimates μ.

Value

a vector of the control points.

See Also

`SplineBasis`

Examples

```
knots<-c(0,0,0,0:5,5,5,5)
base<-SplineBasis(knots)
x<-seq(0,5,by=.5)
y<-exp(x)+rnorm(length(x),sd=5)
fitLS(base,x,y)
```

`GramMatrix` *Computing the Gram Matrix for a set of Spline Basis*

Description

Function for computing the Gram matrix of a spline basis.

Usage

```
GramMatrix(object)
```

Arguments

`object`	a SplineBasis object

Details

Compute the Gram Matrix. If `object` denotes the basis functions $b(t) = \{b_1(t), \ldots, b_J(t)\}$ then the Gram Matrix is,

$$G = \int b^T(t)b(t)\mathrm{d}t$$

Value

a matrix as defined above.

`Hankel` *Generating a Hankel Matrix*

Description

Functions to generate a Hankel matrix.

Usage

```
Hankel(x, nrow = length(x)%/%2, ncol = length(x)%/%2)
```

Arguments

`x`	numeric vector to specify the entries of the matrix. Should have an even number of entries.
`nrow`	integer, must be at most length(x)
`ncol`	integer, must be at most length(x)

Details

Computes a Hankel matrix. If we denote the vector $x = (x_1, \ldots, x_n)$ the Hankel matrix is defined and formed as

$$H = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_{1/2} \\ x_2 & x_3 & & \vdots & \vdots \\ x_3 & & \vdots & & \vdots \\ \vdots & \vdots & & & \vdots \\ x_{1/2} & \cdots & \cdots & \cdots & x_n \end{pmatrix}.$$

Value

a matrix as defined above.

Examples

```
Hankel(1:6)
```

integrate-methods *Methods for Function integrate*

Description

Methods for function `integrate`. `integrate` integrates generic objects for which an integral is defined.

Methods

object = "SplineBasis" Returns a new `SplineBasis` object for the integral of the basis functions. See `SplineBasis`

MatrixPower *Matrix Power*

Description

Performs the matrix power operation.

Usage

```
MatrixPower(A, n)
```

Arguments

`A`	A square matrix.
`n`	An integer telling the exponent.

Details

Only well defined for integers the matrix power operation is a convenience function to multiply a matrix, `A`, with itself `n` times.

Value

A matrix of the same dimension as `A`.

Examples

```
A<-rbind(0,cbind(diag(1:5),0))  #a nilpotent matrix
A
MatrixPower(A,3)
MatrixPower(A,5)
MatrixPower(A,6)    #Gets to a zero matrix
```

orthogonalize-methods *Methods for Function orthogonalize*

Description

A generic function for orthogonalizing an object and returning the orthogonal object

Methods

object = "SplineBasis" Orthogonalize the spline basis functions. See SplineBasis

OrthogonalizeBasis *Orthogonalize a Spline Basis*

Description

Specific function for orthogonalizing the functions in a SplineBasis object.

Usage

```
OrthogonalizeBasis(object, ...)
```

Arguments

object	A SplineBasis object
...	ignored

Value

An OrthogonalSplineBasis object.

See Also

OrthogonalSplineBasis, SplineBasis,orthogonalize

OuterProdSecondDerivative *Outer Product of Second Derivatives of Spline Bases*

Description

Provides the functional outer product of second derivatives of a set of basis functions in a `SplineBasis` object. It a convenient form for forming a penalty on curve smoothness when fitting a spline curve.

Usage

```
OuterProdSecondDerivative(basis)
```

Arguments

`basis`	A `SplineBasis` object

Value

A square matrix of order `nrow(basis)`.

See Also

`SplineBasis`,`fitLS`

SplineBasis *Creating* `SplineBasis` *Objects.*

Description

The function to create `SplineBasis` and `OrthogonalSplineBasis` Objects

Usage

```
SplineBasis(knots, order=4, keep.duplicates=FALSE)
OrthogonalSplineBasis(knots, ...)
OBasis(...)
```

Arguments

`knots`	The full set of knots used to define the basis functions.
`order`	Order of the spline fit.(degree= order-1)
`keep.duplicates`	Should duplicate interior knots that could cause computation problem be kept or removed. Defaults to false, which removes duplicate knots with a warning if duplicate interior knots are found.
`...`	Other arguments either ignored or passed onto other functions.

Details

SplineBasis produces an object representing the basis functions used in spline fitting. Provides a compact easily evaluated representation of the functions. Produces a class of object SplineBasis. OrthogonalSplineBasis is a shortcut to obtain a set of orthogonalized basis functions from the knots. OBasis is an alias for OrthogonalSplineBasis. Both provide an object of class OrthogonalSplineBasis. The class OrthogonalSplineBasis inherits directly from SplineBasis meaning all functions that apply to SplineBasis functions also apply to the orthogonalized version.

Value

Object of class SplineBasis or OrthogonalSplineBasis

References

General matrix representations for B-splines Kaihuai, Qin, The Visual Computer 2000 16:177–186

See Also

SplineBasis, spline, orthogonalsplinebasis-package

Examples

```
knots<-c(0,0,0,0:10,10,10,10)
plot(SplineBasis(knots))
obase<-OBasis(knots)
plot(obase)
dim(obase)[2] #number of functions
evaluate(obase, 1:10-.5)
```

SplineBasis-class *Classes* SplineBasis *and* OrthogonalSplineBasis

---

Description

Contains the matrix representation for spline basis functions. The OrthogonalSplineBasis class has the basis functions orthogonalized.

Objects from the Class

Objects can be created by calls of the form SplineBasis(knots, order) or to generate orthogonal spline basis functions directly OrthogonalSplineBasis(knots, order) or the short version OBasis(knots,order).

Slots

`transformation`: Object of class `"matrix"` Only applicable on `OrthogonalSplineBasis` class, shows the transformation matrix use to get from regular basis functions to orthogonal basis functions.

`knots`: Object of class `"numeric"`

`order`: Object of class `"integer"`

`Matrices`: Object of class `"array"`

Methods

deriv `signature(expr = "SplineBasis")`: Computes the derivative of the basis functions. Returns an object of class `SplineBasis`.

dim `signature(x = "SplineBasis")`: gives the dim as the order and number of basis functions. Returns numeric of length 2.

evaluate `signature(object = "SplineBasis", x = "numeric")`: Evaluates the basis functions and the points provided in `x`. Returns a matrix with `length(x)` rows and `dim(object)[2]` columns.

integrate `signature(object = "SplineBasis")`: computes the integral of the basis functions defined by $\int_{k_0}^{x} b(t)dt$ where k_0 is the first knot. Returns an object of class `SplineBasis`.

orthogonalize `signature(object = "SplineBasis")`: Takes in a `SplineBasis` object, computes the orthogonalization transformation and returns an object of class `OrthogonalSplineBasis`.

plot `signature(x = "SplineBasis", y = "missing")`: Takes an object of class `SplineBasis` and plots the basis functions for the domain defined by the knots in object.

plot `signature(x = "SplineBasis", y = "vector")`: Interprets y as a vector of coefficients and plots the resulting curve.

plot `signature(x = "SplineBasis", y = "matrix")`: Interprets y as a matrix of coefficients and plots the resulting curves.

References

General matrix representations for B-splines Kaihuai Qin, The Visual Computer 2000 16:177–186

See Also

SplineBasis

Examples

```
showClass("SplineBasis")

knots<-c(0,0,0,0:5,5,5,5)
(base <-SplineBasis(knots))
(obase<-OBasis(knots))
plot(base)
plot(obase)
```

Index