Package 'freeknotsplines'

June 10, 2018

Version 1.0.1

Date 2018-05-17

Title Algorithms for Implementing Free-Knot Splines

Author Steven Spiriti <puzzlesteven@gmail.com>, Philip Smith <philip.smith@ttu.edu>, Pierre Lecuyer <lecuyer@iro.UMontreal.ca>

Maintainer Steven Spiriti <puzzlesteven@gmail.com>

Depends R (>= 2.14.2)

Imports splines, methods

Description Algorithms for fitting free-knot splines for data with one independent variable and one dependent variable. Four free-knot spline algorithms are provided for the case where the number of knots is known in advance. A knot-search algorithm is provided for the case where the number of knots is not known in advance. In addition, methods are available to compute the fitted values, the residuals, and the coefficients of the splines, and to plot the results, along with a method to summarize the results.

License GPL (>= 2)

LazyData tsrue

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-06-10 06:04:37 UTC

R topics documented:

`freeknotspline-package` *Free-Knot Splines*

Description

This package is for fitting free-knot splines for data with one independent variable and one dependent variable.

Details

Four fitting methods are included for the case where the number of knots is known in advance; for details, see `freeknotfit`. In addition, methods are available to compute the fitted values, the residuals, and the coefficients of the splines, and to plot the results, along with a method to summarize the results. Finally, a function (see `fit.search.numknots`) is provided to optimize the number of knots using a specified fit criterion. Several fit criteria are provided (see `fitcriteria`).

Author(s)

Steven Spiriti, Philip Smith, and Pierre Lecuyer

References

Spiriti, S., Eubank, R., Smith, P., Young, D., "Knot Selection for Least-Squares and Penalized Splines," *Journal of Statistical Computation and Simulation*, in press.

`chgbasismat` *Compute Change-of-Basis Matrix to Convert Truncated Power Basis to B-Spline Basis*

Description

This function computes a change-of-basis matrix that converts from the truncated power basis to the B-spline basis. It is used by `coef.freekt` and should not be called directly by the user.

Usage

```
chgbasismat(knot, ord)
```

Arguments

`knot`	The vector of knots. The first and last knots are repeated `ord` times to make the length equal to the dimension of the spline space.
`ord`	The order of the spline, which is one more than the degree.

Value

The change-of-basis matrix that converts from the truncated power basis to the B-spline basis.

Author(s)

Steven Spiriti

References

Smith, P. (1982), "Hypothesis Testing B-Spline Regression," *Communications in Statistics, Series B*, 11, 143-157.

Spiriti, S., Eubank, R., Smith, P., Young, D., "Knot Selection for Least-Squares and Penalized Splines," *Journal of Statistical Computation and Simulation*, in press.

`coef.freekt` *Compute Coefficients of B-Splines For Free-Knot Splines*

Description

This function computes the coefficients of the B-splines for free-knot splines, given the amount of the penalty (if applicable) and the locations of the knots.

Usage

```
## S3 method for class 'freekt'
coef(object, ...)
```

Arguments

`object`	An object of class "`freekt`" obtained by using one of the fitting algorithms.
`...`	Additional arguments to be passed to coef.freekt. Currently ignored.

Value

A vector containing the coefficients of the B-splines.

Author(s)

Steven Spiriti

See Also

`fitted.freekt` to compute the fitted values and `residuals.freekt` to compute the residuals.

Examples

```
x <- 0:30/30
truey <- x*sin(10*x)
set.seed(10556)
y <- truey + rnorm(31, 0, 0.2)
xy.freekt <- freelsgen(x, y, degree = 2, numknot = 2, 555)
coef.freekt(xy.freekt)
```

`fit.search.numknots` *Perform a Search on the Number of Knots and Fit Free-Knot Splines To Data Using the Optimal Number of Knots*

Description

This function fits free-knot splines to data using every value for the number of knots between `minknot` and `maxknot`. The number of knots is then chosen to optimize a fit criterion. The free-knot spline with the optimum number of knots is returned.

Usage

```
fit.search.numknots(x, y, degree, minknot = 1, maxknot = 5,
                    alg = "LS", search = "genetic",
                    knotnumcrit = "adjGCV", k = 2, d = 3, seed = 5,
                    stream = 0)
```

Arguments

`x`	A vector containing the values of the independent variable.
`y`	A vector containing the values of the dependent variable.
`degree`	The degree of the spline fit.
`minknot`	The minimum number of knots to search. Defaults to 1.
`maxknot`	The maximum number of knots to search. Defaults to 5.
`alg`	The spline-fitting algorithm. Choices are "LS" for least-squares and "PS" for P-splines. Defaults to "LS."
`search`	The random search algorithm. Choices are "genetic" for a genetic algorithm and "golden" for a blind random search with golden section adjustment. Defaults to "genetic."
`knotnumcrit`	The criterion to be used for determining the number of knots. Choices are "GCV" for generalized cross-validation, "AIC" for the Akaike information criterion, "AICc" for corrected Akaike information criterion, "BIC" for Bayesian information criterion, "adjAIC" for an adjusted version of the Akaiki information criterion, and "adjGCV" for an adjusted version of generalized cross-validation. Defaults to "adjGCV."
`k`	The amount of penalty when AIC is used. Has no effect with criteria other than AIC. Defaults to 2.

d	The amount of penalty when adjGCV is used. Has no effect with criteria other than adjGCV. Defaults to 3.
seed	The value of the initial seed. Defaults to 5.
stream	The value of the initial stream to be used for parallel programming. Defaults to 0.

Value

An object of class "freekt" containing the following components:

x	A vector containing the x values.
y	A vector containing the y values.
degree	The degree of the spline fit.
seed	The value of the initial seed.
stream	The value of the stream.
lambda	The optimum amount of penalty. This is automatically equal to 0 for freelsgen and freelsgold.
optknot	A vector containing the optimal knots.
tracehat	The trace of the hat matrix for the optimal fit.
GCV	The value of generalized cross validation (GCV) for the optimal fit.)
GSJS	The GSJS estimator, an estimator of the variance of the data.
call	The function call.

Author(s)

Steven Spiriti

References

Eubank, R. (1999), *Nonparametric Regression and Spline Smoothing*, New York: Marcel Dekker, Inc., Second ed.

Spiriti, S., Eubank, R., Smith, P., Young, D., "Knot Selection for Least-Squares and Penalized Splines," *Journal of Statistical Computation and Simulation*, in press.

See Also

fitcriteria for the fit criteria, freeknotfit for the free-knot spline algorithms.

Examples

```
x <- 0:30/30
truey <- x*sin(10*x)
set.seed(10556)
y <- truey + rnorm(31, 0, 0.2)
xy.freekt <- fit.search.numknots(x, y, degree = 2, minknot = 1, maxknot = 3, seed = 555)
plot.freekt(xy.freekt, xfit = 0:1000/1000)
```

`fitcriteria` *Fit Criteria for Free-Knot Splines*

Description

These functions compute various criteria for determining the fit of a free-knot spline. `AIC.freekt` computes the Akaike Information Criterion, with `k` determining the amount of the penalty. `AICc.freekt` computes the corrected Akaike Information Criterion. `BIC.freekt` computes the Bayesian Information Criterion, also known as Schwarz Information Criterion. `adjAIC.freekt` computes an adjusted Akaike Information Criterion with the penalty increased to account for the greater flexibility of free knots. `adjGCV.freekt` computes an adjusted GCV with the degrees of freedom increased to account for the greater flexibility of free knots.

Usage

```
## S3 method for class 'freekt'
AIC(object, ..., k = 2)
AICc.freekt(object)
## S3 method for class 'freekt'
BIC(object, ...)
adjAIC.freekt(object)
adjGCV.freekt(object, d = 3)
```

Arguments

`object`	An object of class "`freekt`" obtained by using one of the fitting algorithms.
`k`	The amount of the penalty. Used only for `AIC.freekt`.
`d`	The amount of the penalty. Used only for `adjGCV.freekt`.
`...`	Additional arguments to be passed to the `AIC.freekt` and `BIC.freekt` functions.

Value

Returns the value of the specified fit criterion.

Author(s)

Steven Spiriti

References

Spiriti, S., Eubank, R., Smith, P., Young, D., "Knot Selection for Least-Squares and Penalized Splines," *Journal of Statistical Computation and Simulation*, in press.

See Also

`fit.search.numknots`, which uses these fit criteria to determine the number of knots.

`fitted.freekt` *Compute Fitted Values For Free-Knot Spline*

Description

This function computes the fitted values, given the amount of the penalty (if applicable) and the locations of the knots.

Usage

```
## S3 method for class 'freekt'
fitted(object, xfit = object@x, ...)
```

Arguments

`object`	An object of class "`freekt`" obtained by using one of the fitting algorithms.
`xfit`	A vector of x values at which to compute the fitted values. Defaults to the x values of the data.
`...`	Additional arguments to be passed to fitted.freekt. Currently ignored.

Value

A vector containing the fitted values.

Author(s)

Steven Spiriti

See Also

`residuals.freekt` for the residuals.

Examples

```
x <- 0:30/30
truey <- x*sin(10*x)
set.seed(10556)
y <- truey + rnorm(31, 0, 0.2)
xy.freekt <- freelsgen(x, y, degree = 2, numknot = 2, 555)
fitted.freekt(xy.freekt)
```

freeknotfit *Fit Free-Knot Splines To Data*

Description

These functions fit free-knot splines to data with one independent variable and one dependent variable. It is assumed that the number of knots is known in advance. `freelsgen` and `freelsgold` fit least-squares splines with no penalty, while `freepsgen` and `freepsgold` fit penalized splines. `freelsgen` and `freepsgen` use a genetic algorithm, while `freelsgold` and `freepsgold` use a blind search augmented with a golden section algorithm.

Usage

```
freelsgen(x, y, degree, numknot, seed = 5, stream = 0)
freelsgold(x, y, degree, numknot, seed = 5, stream = 0)
freepsgen(x, y, degree, numknot, seed = 5, stream = 0)
freepsgold(x, y, degree, numknot, seed = 5, stream = 0)
```

Arguments

`x`	A vector containing the values of the independent variable.
`y`	A vector containing the values of the dependent variable.
`degree`	The degree of the spline fit.
`numknot`	The number of knots.
`seed`	The value of the initial seed. Defaults to 5.
`stream`	The value of the initial stream to be used for parallel programming. Defaults to 0.

Value

An object of class "`freekt`" containing the following components:

`x`	A vector containing the x values.
`y`	A vector containing the y values.
`degree`	The degree of the spline fit.
`seed`	The value of the initial seed.
`stream`	The value of the stream.
`lambda`	The optimum amount of penalty. This is automatically equal to 0 for `freelsgen` and `freelsgold`.
`optknot`	A vector containing the optimal knots.
`tracehat`	The trace of the hat matrix for the optimal fit.
`GCV`	The value of generalized cross validation (GCV) for the optimal fit.
`GSJS`	The GSJS estimator, an estimator of the variance of the data.
`call`	The function call.

Author(s)

Steven Spiriti, Philip Smith, and Pierre Lecuyer

References

Eubank, R. (1999), *Nonparametric Regression and Spline Smoothing*, New York: Marcel Dekker, Inc., Second ed.

Spiriti, S., Eubank, R., Smith, P., Young, D., "Knot Selection for Least-Squares and Penalized Splines," *Journal of Statistical Computation and Simulation*, in press.

See Also

`fit.search.numknots` for the case where the number of knots is not specified in advance.

Examples

```
x <- 0:30/30
truey <- x*sin(10*x)
set.seed(10556)
y <- truey + rnorm(31, 0, 0.2)
xy.freekt <- freelsgen(x, y, degree = 2, numknot = 2, 555)
plot.freekt(xy.freekt, xfit = 0:1000/1000)
```

---

plot.freekt *Plot Fitted Values for Free-Knot Spline*

Description

This function plots the fit obtained using a free-knot spline.

Usage

```
## S3 method for class 'freekt'
plot(x, xfit = x@x, linecolor = "blue", lwd = 1, lty = 1, ...)
```

Arguments

`x`	An object of class "`freekt`" obtained by using one of the fitting algorithms.
`xfit`	A vector of x values at which to plot the fitted values. Defaults to the x values of the data.
`linecolor`	The color of the line. Defaults to blue.
`lwd`	The line width. It is passed to the `lines` function. Defaults to 1.
`lty`	The line type. It is passed to the `lines` function. Defaults to 1.
`...`	Additional arguments to be passed to the `plot` function.

Value

A plot of the data, together with the spline estimator.

Author(s)

Steven Spiriti

See Also

`fitted.freekt` to compute the fitted values.

Examples

```
x <- 0:30/30
truey <- x*sin(10*x)
set.seed(10556)
y <- truey + rnorm(31, 0, 0.2)
xy.freekt <- freelsgen(x, y, degree = 2, numknot = 2, 555)
plot.freekt(xy.freekt, xfit = 0:1000/1000)
```

---

residuals.freekt *Compute Residuals For Free-Knot Spline*

Description

This function computes the residuals, given the optimal values for lambda and the locations of the knots.

Usage

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

Arguments

`object`	An object of class "`freekt`" obtained by using one of the fitting algorithms.
`...`	Additional arguments to be passed to residuals.freekt. Currently ignored.

Value

A vector containing the residuals.

Author(s)

Steven Spiriti

See Also

`fitted.freekt` to compute the fitted values.

Examples

```
x <- 0:30/30
truey <- x*sin(10*x)
set.seed(10556)
y <- truey + rnorm(31, 0, 0.2)
xy.freekt <- freelsgen(x, y, degree = 2, numknot = 2, 555)
plot(x, residuals(xy.freekt))
```

---

summary.freekt *Summarize Free-Knot Spline Fit*

Description

This function displays a summary of the fit obtained using a free-knot spline.

Usage

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

Arguments

object	An object of class "freekt" obtained by using one of the fitting algorithms.
...	Additional arguments to be passed to summary.freekt. Currently ignored.

Value

A table containing the values of the optimal amount of penalty (when applicable), the optimal knots, sum of squared errors (SSE), and generalized cross-validation (GCV).

Author(s)

Steven Spiriti

See Also

freeknotfit for the fitting algorithms.

Examples

```
x <- 0:30/30
truey <- x*sin(10*x)
set.seed(10556)
y <- truey + rnorm(31, 0, 0.2)
xy.freekt <- freelsgen(x, y, degree = 2, numknot = 2, 555)
summary.freekt(xy.freekt)
```

Index