Package 'SiZer'

February 13, 2020

Version 0.1-7

Date 2020-2-12

Title Significant Zero Crossings

Depends R (>= 2.4.0)

Imports stats, graphics, splines, boot

Suggests dplyr, ggplot2

Description Calculates and plots the SiZer map for scatterplot data. A SiZer map is a way of examining when the p-th derivative of a scatterplot-smoother is significantly negative, possibly zero or significantly positive across a range of smoothing bandwidths.

License GPL (>= 2)

URL https://github.com/dereksonderegger/SiZer

RoxygenNote 7.0.2

NeedsCompilation no

Author Derek Sonderegger [aut, cre]

Maintainer Derek Sonderegger <derek.sonderegger@gmail.com>

Repository CRAN

Date/Publication 2020-02-13 06:20:14 UTC

R topics documented:

Arkansas 2
bent.cable 3
locally.weighted.polynomial 4
logLik.bent_cable 6
logLik.PiecewiseLinear 7
piecewise.linear 7
plot.LocallyWeightedPolynomial 9
plot.PiecewiseLinear 9
plot.SiZer 10
predict.bent_cable 11

predict.PiecewiseLinear 12
print.PiecewiseLinear 12
SiZer 13

Index **15**

Arkansas *Time Series of Macroinvertabrates Abundance in the Arkansas River.*

Description

A time series of 16 years (5 replicates per year) of mayfly (Ephemeroptera:Heptageniidae) abundance in the fall at the monitoring station AR1 on the Arkansas River in Colorado, USA.

Usage

```
data(Arkansas, package='SiZer')
```

Format

A data frame with 90 observations on the following 2 variables.

year The year of observation

sqrt.mayflies The Square root of observed abundance.

Source

Sonderegger, D.L., Wang, H., Clements, W.H., and Noon, B.R. 2009. Using SiZer to detect thresholds in ecological data. Frontiers in Ecology and the Environment 7:190-195.

Examples

```
require(ggplot2)

data(Arkansas)
ggplot(Arkansas, aes(x=year, y=sqrt.mayflies)) +
   geom_point()
```

`bent.cable` *Fits a bent-cable model to the given data Fits a bent-cable model to the given data by exhaustively searching the 2-dimensional parameter space to find the maximum likelihood estimators for α and γ.*

Description

Fits a bent-cable model to the given data Fits a bent-cable model to the given data by exhaustively searching the 2-dimensional parameter space to find the maximum likelihood estimators for α and γ.

Usage

```
bent.cable(x, y, grid.size = 100)
```

Arguments

`x`	The independent variable
`y`	The dependent variable
`grid.size`	How many α and $gamma$ values to examine. The total number of parameter combinations examined is `grid.size` squared.

Details

Fit the model which is essentially a piecewise linear model with a quadratic curve of length 2γ connecting the two linear pieces.

The reason for searching the space exhaustively is because the bent-cable model often has a likelihood surface with a very flat ridge instead of definite peak. While the exhaustive search is slow, at least it is possible to examine the contour plot of the likelihood surface.

@return A list of 7 elements:

log.likelihood A matrix of log-likelihood values.

SSE A matrix of sum-of-square-error values.

alphas A vector of alpha values examined.

gammas A vector of gamma values examined.

alpha The MLE estimate of alpha.

gamma The MLE estimate of gamma.

model The `lm` fit after $alpha$ and $gamma$ are known.

Author(s)

Derek Sonderegger

References

Chiu, G. S., R. Lockhart, and R. Routledge. 2006. Bent-cable regression theory and applications. Journal of the American Statistical Association 101:542-553.

Toms, J. D., and M. L. Lesperance. 2003. Piecewise regression: a tool for identifying ecological thresholds. Ecology 84:2034-2041.

See Also

piecewise.linear

Examples

```
data(Arkansas)
x <- Arkansas$year
y <- Arkansas$sqrt.mayflies

# For a more accurate estimate, increase grid.size
model <- bent.cable(x,y, grid.size=20)
plot(x,y)
x.grid <- seq(min(x), max(x), length=200)
lines(x.grid, predict(model, x.grid), col='red')
```

---

locally.weighted.polynomial

Smoothes the given bivariate data using kernel regression.

Description

Smoothes the given bivariate data using kernel regression.

Usage

```
locally.weighted.polynomial(
  x,
  y,
  h = NA,
  x.grid = NA,
  degree = 1,
  kernel.type = "Normal"
)
```

Arguments

x	Vector of data for the independent variable
y	Vector of data for the dependent variable
h	The bandwidth for the kernel

`x.grid`	What x-values should the value of the smoother be calculated at.
`degree`	The degree of the polynomial to be fit at each x-value. The default is to fit a linear regression, ie degree=1.
`kernel.type`	What kernel to use. Valid choices are 'Normal', 'Epanechnikov', 'biweight', and 'triweight'.

Details

The confidence intervals are created using the row-wise method of Hannig and Marron (2006).

Notice that the derivative to be estimated must be less than or equal to the degree of the polynomial initially fit to the data.

If the bandwidth is not given, the Sheather-Jones bandwidth selection method is used.

Value

Returns a `LocallyWeightedPolynomial` object that has the following elements:

data A structure of the data used to generate the smoothing curve

h The bandwidth used to generate the smoothing curve.

x.grid The grid of x-values that we have estimated function value and derivative(s) for.

degrees.freedom The effective sample size at each grid point

Beta A matrix of estimated beta values. The number of rows is degrees+1, while the number of columns is the same as the length of x.grid. Notice that

$$\hat{f}(x_i) = \beta[1, i]$$

$$\hat{f}'(x_i) = \beta[2, i] * 1!$$

$$\hat{f}''(x_i) = \beta[3, i] * 2!$$

and so on...

Beta.var Matrix of estimated variances for `Beta`. Same structure as `Beta`.

Author(s)

Derek Sonderegger

References

Chaudhuri, P., and J. S. Marron. 1999. SiZer for exploration of structures in curves. Journal of the American Statistical Association 94 807-823.

Hannig, J., and J. S. Marron. 2006. Advanced distribution theory for SiZer. Journal of the American Statistical Association 101 484-499.

Sonderegger, D.L., Wang, H., Clements, W.H., and Noon, B.R. 2009. Using SiZer to detect thresholds in ecological data. Frontiers in Ecology and the Environment 7:190-195

See Also

SiZer, plot.LocallyWeightedPolynomial, spm in package 'SemiPar', loess, smooth.spline, interpSpline in the splines package.

Examples

```
data(Arkansas)
x <- Arkansas$year
y <- Arkansas$sqrt.mayflies
layout(cbind(1,2,3))
model <- locally.weighted.polynomial(x,y)
plot(model, main='Smoothed Function', xlab='Year', ylab='Sqrt.Mayflies')

model2 <- locally.weighted.polynomial(x,y,h=.5)
plot(model2, main='Smoothed Function', xlab='Year', ylab='Sqrt.Mayflies')

model3 <- locally.weighted.polynomial(x,y, degree=1)
plot(model3, derv=1, main='First Derivative', xlab='Year', ylab='1st Derivative')
```

---

logLik.bent_cable *Return the log-Likelihood value for a fitted bent-cable model.*

Description

Return the log-Likelihood value for a fitted bent-cable model.

Usage

```
## S3 method for class 'bent_cable'
logLik(object, ...)
```

Arguments

object	A bent-cable model
...	Unused at this time.

`logLik.PiecewiseLinear` *Calculates the log-Likelihood value*

Description

Calculates the log-Likelihood value

Usage

```
## S3 method for class 'PiecewiseLinear'
logLik(object, ...)
```

Arguments

`object`	A `PiecewiseLinear` object
`...`	Unused at this time.

`piecewise.linear` *Creates a piecewise linear model*

Description

Fit a degree 1 spline with 1 knot point where the location of the knot point is unknown.

Usage

```
piecewise.linear(
  x,
  y,
  middle = 1,
  CI = FALSE,
  bootstrap.samples = 1000,
  sig.level = 0.05
)
```

Arguments

`x`	Vector of data for the x-axis.
`y`	Vector of data for the y-axis
`middle`	A scalar in $[0, 1]$. This represents the range that the change-point can occur in. 0 means the change-point must occur at the middle of the range of x-values. 1 means that the change-point can occur anywhere along the range of the x-values.

`CI` Whether or not a bootstrap confidence interval should be calculated. Defaults to FALSE because the interval takes a non-trivial amount of time to calculate

`bootstrap.samples` The number of bootstrap samples to take when calculating the CI.

`sig.level` What significance level to use for the confidence intervals.

Details

The bootstrap samples are taken by resampling the raw data points. Sometimes a more appropriate bootstrap sample would be to calculate the residuals and then add a randomly selected residual to each y-value.

Value

A list of 5 elements is returned:

change.point The estimate of α.

model The resulting `lm` object once α is known.

x The x-values used.

y The y-values used.

CI Whether or not the confidence interval was calculated.

intervals If the CIs where calculated, this is a matrix of the upper and lower intervals.

References

Chiu, G. S., R. Lockhart, and R. Routledge. 2006. Bent-cable regression theory and applications. Journal of the American Statistical Association 101:542-553.

Toms, J. D., and M. L. Lesperance. 2003. Piecewise regression: a tool for identifying ecological thresholds. Ecology 84:2034-2041.

See Also

The package `segmented` has a much more general implementation of this analysis and users should preferentially use that package.

Examples

```
data(Arkansas)
x <- Arkansas$year
y <- Arkansas$sqrt.mayflies

model <- piecewise.linear(x,y, CI=FALSE)
plot(model)
print(model)
predict(model, 2001)
```

`plot.LocallyWeightedPolynomial` *Creates a plot of an object created by* `locally.weighted.polynomial`.

Description

Creates a plot of an object created by `locally.weighted.polynomial`.

Usage

```
## S3 method for class 'LocallyWeightedPolynomial'
plot(
  x,
  derv = 0,
  CI.method = 2,
  alpha = 0.05,
  use.ess = TRUE,
  draw.points = TRUE,
  ...
)
```

Arguments

`x`	LocallyWeightedPolynomial object
`derv`	Derivative to be plotted. Default is 0 - which plots the smoothed function.
`CI.method`	What method should be used to calculate the confidence interval about the estimated line. The methods are from Hannig and Marron (2006), where 1 is the point-wise estimate, and 2 is the row-wise estimate.
`alpha`	The alpha level such that the CI has a 1-alpha/2 level of significance.
`use.ess`	ESS stands for the estimated sample size. If at any point along the x-axis, the ESS is too small, then we will not plot unless use.ess=FALSE.
`draw.points`	Should the data points be included in the graph? Defaults to TRUE.
`...`	Additional arguments to be passed to the graphing functions.

`plot.PiecewiseLinear` *Plots a piecewise linear model*

Description

Plots a piecewise linear model

Usage

```
## S3 method for class 'PiecewiseLinear'
plot(x, xlab = "X", ylab = "Y", ...)
```

Arguments

x	A PiecewiseLinear object
xlab	The label for the x-axis
ylab	The label for the y-axis
...	Any further options to be passed to the plot function

---

plot.SiZer   *Plot a SiZer map Plot a* SiZer *object that was created using* SiZer()

Description

Plot a SiZer map Plot a SiZer object that was created using SiZer()

Usage

```
## S3 method for class 'SiZer'
plot(
  x,
  ylab = expression(log[10](h)),
  colorlist = c("red", "purple", "blue", "grey"),
  ...
)
```

Arguments

x	An object created using SiZer()
ylab	What the y-axis should be labled.
colorlist	What colors should be used. This is a vector that corresponds to 'decreasing', 'possibley zero', 'increasing', and 'insufficient data'.
...	Any other parameters to be passed to the function image.

Details

The white lines in the SiZer map give a graphical representation of the bandwidth. The horizontal distance between the lines is $2h$.

Author(s)

Derek Sonderegger

References

Chaudhuri, P., and J. S. Marron. 1999. SiZer for exploration of structures in curves. Journal of the American Statistical Association 94:807-823.

Hannig, J., and J. S. Marron. 2006. Advanced distribution theory for SiZer. Journal of the American Statistical Association 101:484-499.

Sonderegger, D.L., Wang, H., Clements, W.H., and Noon, B.R. 2009. Using SiZer to detect thresholds in ecological data. Frontiers in Ecology and the Environment 7:190-195.

See Also

plot.SiZer, locally.weighted.polynomial

Examples

```
data('Arkansas')
x <- Arkansas$year
y <- Arkansas$sqrt.mayflies

plot(x,y)

# Calculate the SiZer map for the first derivative
SiZer.1 <- SiZer(x, y, h=c(.5,10), degree=1, derv=1, grid.length=21)
plot(SiZer.1)

# Calculate the SiZer map for the second derivative
SiZer.2 <- SiZer(x, y, h=c(.5,10), degree=2, derv=2, grid.length=21);
plot(SiZer.2)

# By setting the grid.length larger, we get a more detailed SiZer
# map but it takes longer to compute.
#
# SiZer.3 <- SiZer(x, y, h=c(.5,10), grid.length=100, degree=1, derv=1)
# plot(SiZer.3)
```

---

predict.bent_cable *Return model predictions for fitted bent-cable model*

Description

Return model predictions for fitted bent-cable model

Usage

```
## S3 method for class 'bent_cable'
predict(object, x, ...)
```

Arguments

`object`	A bent-cable model
`x`	The set x-values for which predictions are desired
`...`	A placeholder that is currently ignored.

`predict.PiecewiseLinear` *Calculates predicted values from a piecewise linear object*

Description

Calculates predicted values from a piecewise linear object

Usage

```
## S3 method for class 'PiecewiseLinear'
predict(object, x, ...)
```

Arguments

`object`	A `PiecewiseLinear` object
`x`	A vector of x-values in which to calculate the y.
`...`	Unused at this time.

`print.PiecewiseLinear` *Prints out the model form for a Piecewise linear model*

Description

Prints out the model form for a Piecewise linear model

Usage

```
## S3 method for class 'PiecewiseLinear'
print(x, ...)
```

Arguments

`x`	A `PiecewiseLinear` object
`...`	Unused at this time.

`SiZer` *Calculate SiZer Map*

Description

Calculates the SiZer map from a given set of X and Y variables.

Usage

```
SiZer(x, y, h=NA, x.grid=NA, degree=NA, derv=1, grid.length=41)
```

Arguments

`x`	data vector for the independent axis
`y`	data vector for the dependent axis
`h`	An integer representing how many bandwidths should be considered, or vector of length 2 representing the upper and lower limits h should take, or a vector of length greater than two indicating which bandwidths to examine.
`x.grid`	An integer representing how many bins to use along the x-axis, or a vector of length 2 representing the upper and lower limits the x-axis should take, or a vector of length greater than two indicating which x-values the derivative should be evaluated at
`degree`	The degree of the local weighted polynomial used to smooth the data. This must be greater than or equal to `derv`.
`derv`	The order of derivative for which to make the SiZer map.
`grid.length`	The default length of the `h.grid` or `x.grid` if the length of either is not given.

Details

SiZer stands for the Significant Zero crossings of the derivative. There are two dominate approaches in smoothing bivariate data: locally weighted regression or penalized splines. Both approaches require the use of a 'bandwidth' parameter that controls how much smoothing should be done. Unfortunately there is no uniformly best bandwidth selection procedure. SiZer (Chaudhuri and Marron, 1999) is a procedure that looks across a range of bandwidths and classifies the p-th derivative of the smoother into one of three states: significantly increasing (blue), possibly zero (purple), or significantly negative (red).

Value

Returns list object of type SiZer which has the following components:

x.grid Vector of x-values at which the derivative was evaluated.

h.grid Vector of bandwidth values for which a smoothing function was calculated.

slopes Matrix of what category a particular x-value and bandwidth falls into (Increasing=1, Possibly Zero=0, Decreasing=-1, Not Enough Data=2).

Author(s)

Derek Sonderegger

References

Chaudhuri, P., and J. S. Marron. 1999. SiZer for exploration of structures in curves. Journal of the American Statistical Association 94:807-823.

Hannig, J., and J. S. Marron. 2006. Advanced distribution theory for SiZer. Journal of the American Statistical Association 101:484-499.

Sonderegger, D.L., Wang, H., Clements, W.H., and Noon, B.R. 2009. Using SiZer to detect thresholds in ecological data. Frontiers in Ecology and the Environment 7:190-195.

See Also

plot.SiZer, locally.weighted.polynomial

Examples

```
data('Arkansas')
x <- Arkansas$year
y <- Arkansas$sqrt.mayflies

plot(x,y)

# Calculate the SiZer map for the first derivative
SiZer.1 <- SiZer(x, y, h=c(.5,10), degree=1, derv=1, grid.length=21)
plot(SiZer.1)

# Calculate the SiZer map for the second derivative
SiZer.2 <- SiZer(x, y, h=c(.5,10), degree=2, derv=2, grid.length=21);
plot(SiZer.2)

# By setting the grid.length larger, we get a more detailed SiZer
# map but it takes longer to compute.
#
# SiZer.3 <- SiZer(x, y, h=c(.5,10), grid.length=100, degree=1, derv=1)
# plot(SiZer.3)
```

Index

∗Topic **datasets**
    Arkansas, 2

Arkansas, 2

bent.cable, 3

interpSpline, *6*

locally.weighted.polynomial, 4, *11*, *14*
loess, *6*
logLik.bent_cable, 6
logLik.PiecewiseLinear, 7

piecewise.linear, *4*, 7
plot.LocallyWeightedPolynomial, *6*, 9
plot.PiecewiseLinear, 9
plot.SiZer, 10, *11*, *14*
predict.bent_cable, 11
predict.PiecewiseLinear, 12
print.PiecewiseLinear, 12

SiZer, *6*, 13
smooth.spline, *6*