Package ‘iosmooth’

February 20, 2015

Type Package
Title Functions for smoothing with infinite order flat-top kernels
Version 0.91
Date 2014-08-19
Author Timothy L. McMurry, Dimitris N. Politis
Maintainer Timothy L. McMurry <tmcmurry@virginia.edu>
Description Density, spectral density, and regression estimation using infinite order kernels.
License GPL
NeedsCompilation no
Repository CRAN
Date/Publication 2014-08-29 15:06:26

R topics documented:

bwadap ................................................. 2
bwadap.numeric ........................................ 2
bwadap.ts ............................................. 4
bwplot ................................................. 5
bwplot.numeric ........................................ 6
bwplot.ts ............................................. 7
iodensity ............................................. 8
ioksmooth ............................................ 10
iospecden ............................................ 11
sospecden ............................................ 12

Index 14
bwadap

Adaptive bandwidth choice

Description

Generic function for adaptive bandwidth choice.

Usage

bwadap(x, ...)

Arguments

x If x is a univariate time series with class ts, bwadap returns an appropriate
smoothing parameter for spectral density estimation. If x is a univariate data
vector with class numeric, bwadap returns a bandwidth for density estimation.

... Further arguments passed to the methods.

Author(s)

Timothy L. McMurry

References

517-533.

See Also

bwadapnumeric,bwadap.ts

bwadapnumeric

Adaptive bandwidth choice for infinite order density estimates

Description

Adaptive bandwidth choice for infinite order flat-top kernel density estimates.

Usage

## S3 method for class 'numeric'
bwadap(x, smax = 13.49/IQR(x), n.points = 1000, Kn = 1.349 *
5/IQR(x), c.thresh = 2, ...)

bwadapnumeric
Arguments

x  A univariate data set.
smax  The algorithm searches for smoothing parameters on the interval [0, smax]. smax is problem dependent, and the defaults are not consistently appropriate.
n.points  The number of points in [0, smax] at which the algorithm looks for crossing of the threshold c.thresh*sqrt(log(n,10))/n. If the sample characteristic function is highly oscillatory on [0,smax], this may need to be increased.
Kn  Tuning parameter Kn discussed in Politis (2003). Roughly, the distance over which the sample characteristic function must stay below the threshold determined by c.thresh.
c.thresh  The bandwidth is chosen by looking for the first time the sample characteristic function drops below c.thresh*sqrt(log(n,10))/n and stays below that level for a distance of Kn.
...  Currently unimplemented.

Details

Returns a bandwidth, h, for use with infinite order flat-top kernel density estimates. All frequencies higher than 1/h are downweighted by the kernel. All kernels in this package are scaled to use roughly the same bandwidth. We recommend using this algorithm in conjunction with bwplot.numeric to double check the automated selection.

Value

Returns the estimated kernel bandwidth h.

Author(s)

Timothy L. McMurry

References


See Also

bwadap, bwadap.ts, bwplot.numeric, bwplot

Examples

```r
x <- rnorm(100)
bwplot(x)
h <- bwadap(x)
plot(iodensity(x, h, kernel="Trap"), type="l")
rug(x)
# Add the truth in red
xs <- seq(-3, 3, len=1000)
lines(xs, dnorm(xs), col="red")
```
Description

Adaptive bandwidth choice for spectral density estimation with infinite order flat-top kernels.

Usage

```r
## S3 method for class 'ts'
bwadap(x, Kn = 5, c.thresh = 2, ...)
```

Arguments

- `x`: A univariate time-series.
- `Kn`: Tuning parameter Kn discussed in Politis (2003). Roughly, the number of lags the autocorrelation function must stay below the threshold determined by `c.thresh`.
- `c.thresh`: The bandwidth is chosen by looking for the first time the autocorrelation function drops below \( c \cdot \text{thresh} \cdot \sqrt{\log(n)} / n \) and stays below that level for `Kn` lags.
- `...`: Currently unimplemented.

Details

Returns a smoothing parameter 1; all lags after lag 1 will be down-weighted by the kernel’s taper. All kernels in this package are scaled to use roughly the same smoothing parameter.

Value

Smoothing parameter 1.

Author(s)

Timothy L. McMurry

References


See Also

`bwadap, bwadap, bwplot, bwplot.ts`
Examples

```r
x <- arima.sim(list(ar=.7, ma=-.3), 100)
bwplot(x)
l <- bwadap(x)
plot(iospecden(x, l), type="l")
```

bwplot

**Bandwidth plot**

Description

Generic function plotting either the absolute autocorrelation or characteristic function along with the threshold used for bandwidth choice.

Usage

```r
bwplot(x, ...)
```

Arguments

- `x`  
  If `x` is a univariate time series with class `ts`, `bwadap` returns an appropriate smoothing parameter for spectral density estimation. If `x` is a univariate data vector with class `numeric`, `bwadap` returns a bandwidth for density estimation.

- `...`  
  Further arguments passed to the methods.

Author(s)

Timothy L. McMurry

References


See Also

`bwadap`, `bwplot.numeric`, `bwplot.ts`
bwplot.numeric

Bandwidth plot for density estimation or regression

Description

Plots the magnitude of the empirical characteristic function at frequencies 0 to smax.

Usage

```r
## S3 method for class 'numeric'
bwplot(x, y, smax = 13.49/IQR(x), normalize = FALSE, n.points = 1000, c.thresh = 2, ...)
```

Arguments

- `x`: A univariate data set.
- `y`: A vector of responses, for regression only.
- `smax`: The algorithm searches for smoothing parameters on the interval \([0, \text{smax}]\). smax is problem dependent, and the defaults are not consistently appropriate.
- `normalize`: Rescale the sample characteristic function so that it is 1 at frequency 0. Does not affect plots for density estimation. For regression it is often better to leave this as `FALSE`.
- `n.points`: The number of points in \([0, \text{smax}]\) at which the algorithm looks for crossing of the threshold `c.thresh*sqrt(log(n/IQR)/n)`. If the sample characteristic function is highly oscillatory on \([0, \text{smax}]\), this may need to be increased.
- `c.thresh`: For regression, this is not meaningful. For density estimation, the bandwidth is chosen by looking for the first time the sample characteristic function drops below `c.thresh*sqrt(log(n/IQR)/n)` and stays below that level for a distance of Kn.
- `...`: Currently unimplemented.

Details

Produces a plot that is helpful in choosing the bandwidth for infinite order flat-top kernel smoothers. Roughly, the bandwidth should be chosen to let the large low frequency component pass unpre- turbed while damping out smaller high frequency components. This can be accomplished by choosing \(h = 1/l\) where \(l\) is a frequency threshold above which the sample characteristic function is negligible. For regression, this is done heuristically. For density estimation there is a formal recommendation, implemented in `bwplot.numeric`.

Author(s)

Timothy L. McMurry
References


See Also

bwadap, bwadap.ts, bwplot.numeric, bwplot

Examples

```r
### Density Estimation
set.seed(123)
x <- rnorm(100)
bwplot(x, smax=8)
# Choose bandwidth roughly h=1/2
plot(iodensity(x, bw=1/2), type="l")
rug(x)

### Nadaraya-Watson kernel regression
y <- sin(x) + .1 * rnorm(100)
bwplot(x, y, smax=12)
# Choose bandwidth roughly h = 1/2.5
plot(x, y)
lines(ioksmooth(x, y, bw = 1/2.5, kernel="SupSm"), type="l")
```

bwplot.ts  
*Bandwidth plot for spectral density estimation*

Description

Plots the absolute autocorrelation function at lags 0 to lag.max.

Usage

```r
## S3 method for class 'ts'
bwplot(x, lag.max = NULL, c.thresh = 2, ...)
```

Arguments

- `x`  
  A univariate time series.
- `lag.max`  
  The maximum lag shown on the plot.
- `c.thresh`  
  The smoothing parameter is chosen by looking for the first time the sample autocorrelation function drops below `c.thresh*sqrt(log(n)/n)` and stays below that level for a distance of Kn.
- `...`  
  Further arguments passed on to `plot.acf`.
iodensity

Details

Produces a plot that is helpful in choosing the bandwidth for infinite order flat-top spectral density estimates. The smoothing parameter should be chosen to let the large small lag autocorrelations pass unpreturbed while damping out smaller higher lag correlations.

Author(s)

Timothy L. McMurry

References


See Also

`bwadap, bwadap.ts, bwplot, bwplot.numeric`

Examples

```r
goingu::set.seed(123)

x <- arima.sim(list(ar=.7, ma=-.3), 100)
bwplot(x)
bwadap(x)

# Choose a smoothing parameter of 3
plot(iospecden(x, l=3), type="l")
```

iodensity

Kernel density estimation with infinite order kernels

Description

Calculates the standard kernel density estimate using infinite order flat-top kernels. These estimators have been shown to automatically achieve optimal rates of convergence across a wide range of scenarios.

Usage

```r
iodensity(x, bw, kernel = c("Trap", "Rect", "SupSm"), n.points = 100, x.points)
```

Arguments

- **x**: Univariate data x.
- **bw**: The kernel bandwidth. See `bwplot` and `bwadap`.
Three flat-top kernels are implemented, described by the shape of their Fourier transforms. "Trap" is trapezoid shaped and is the default. The rectangular kernel is not recommended and is here for comparison only. SupSm is infinitely differentiable in the Fourier domain; its inverse Fourier transform is estimated numerically, and will be slower.

The number of points at which the density estimate will be calculated if x.points is not specified.

The points at which the density should be calculated. If missing, the function defaults to the range of x +/- 5%.

A list of length 2

The x values at which the density is estimated (x.points if specified).

The estimated density at the associated x values.

Timothy L. McMurry


bwadap, bwadap.numeric
ioksmooth: Nadaraya-Watson regression estimator with infinite order kernels.

Description

Calculates the Nadaraya-Watson nonparametric kernel regression estimator using infinite order flat-top kernels. These estimators have been shown to automatically achieve optimal rates of convergence across a wide range of scenarios.

Usage

ioksmooth(x, y, bw = 0.1, kernel = c("Trap", "Rect", "SupSm"), n.points = 100, x.points)

Arguments

- **x**: Predictor variable x.
- **y**: Response variable y.
- **bw**: The kernel bandwidth. Can be chosen with the help of bwplot.
- **kernel**: Three flat-top kernels are implemented, described by the shape of their Fourier transforms. "Trap" is trapezoid shaped and is the default. The rectangular kernel is not recommended and is here for comparison only. SuperSmooth is infinitely differentiable in the Fourier domain, which, in theory, minimizes edge effects in the interior of the domain. It is calculated numerically, and will be slower.
- **n.points**: The number of points at which the density estimate will be calculated if x.points is not specified.
- **x.points**: The points at which the density should be calculated. If missing, the function defaults to the range of x +/- 5%.

Value

A list of length 2

- **x**: The x values at which the regression is estimated (x.points if specified).
- **y**: The estimated regression function at the associated x values.

Warning

The regressions can be unstable in regions where the design density is low. For regression, bwplot can be used to visually select the large low frequency component, but it is not amenable to algorithmic bandwidth choice.

Author(s)

Timothy L. McMurry
iospecden

Spectral density estimation with infinite order kernels

Description
Calculates a spectral density estimator using infinite order flat-top kernels. These estimators have been shown to automatically achieve optimal rates of convergence across a wide range of scenarios.

Usage
iospecden(x, l, kernel = c("Trap", "Rect", "SupSm"), x.points = seq(-pi, pi, len = 200))

Arguments
x A univariate time series.
l The smoothing parameter. If missing adaptive bandwidth choice is used via bwadap.ts.
kernel Three flat-top kernels are implemented, described by the shape of their Fourier transforms. "Trap" is trapezoid shaped and is the default. The rectangular kernel is not recommended and is here for comparison only. SupSm is infinitely differentiable in the Fourier domain.
x.points Points at which the spectral density is estimated. If x.points is set to NULL, iospecden will return a function which interpolates between the estimated points.

Examples
```r
set.seed(123)
x <- sort(runif(200, 0, 2*pi))
# Regression function
r <- exp(-x^2) + sin(x)
# Observed response
y <- r + 0.3*runif(200)
bwplot(x, y, smax=10)
# Choose bandwidth about 1/2
plot(x, y)
lines(ioksmooth(x, y, bw=1/2, kernel="Trap"))
# Add the truth in red
lines(x, r, col="red")
```

See Also
bwplot, bwplot.numeric

References

Value

If `x.points` is not NULL, the function returns a list of length 2

- `x` - The x values at which the spectral density is estimated (`x.points` if specified).
- `y` - The estimated spectral density function at the associated x values.

If `x.points` is NULL, the function returns the estimated spectral density function rather than its values.

Author(s)

Timothy L. McMurry

References


See Also

`bwadap.ts, bwplot.ts`

Examples

```r
x <- arima.sim(list(ar=.7, ma=-.3), 100)
bwplot(x)
plot(iospecden(x), type="l")
```

Description

Calculates a spectral density estimator using Parzen’s piecewise cubic lag window, with plug-in bandwidth chosen using an infinite order pilot.

Usage

```r
sospecden(x, l, kernel = c("Trap", "Rect", "SupSm"), x.points = seq(-pi, pi, len = 200))
```
Arguments

- **x**: A univariate time series.
- **l**: The smoothing parameter used for the infinite order pilot estimate. If missing, adaptive bandwidth choice is used via `bwadap.ts`.
- **kernel**: The flat-top kernel used for the pilot estimate. Three kernels are implemented, described by the shape of their Fourier transforms. "Trap" is trapezoid shaped and is the default. The rectangular kernel is not recommended and is here for comparison only. SupSm is infinitely differentiable in the Fourier domain.
- **x.points**: Points at which the spectral density is estimated. If `x.points` is set to `NULL`, `sospecden` will return a function which interpolates between the estimated points.

Value

If `x.points` is not NULL, the function returns a list of length 2:

- **x**: The x values at which the spectral density is estimated (x.points if specified).
- **y**: The estimated spectral density function at the associated x values.

If `x.points` is NULL, the function returns the estimated spectral density function rather than its values.

Author(s)

Timothy L. McMurry

References


See Also

`bwadap.ts`, `bwplot.ts`

Examples

```r
x <- arima.sim(list(ar=.7, ma=-.3), 100)
bwplot(x)
plot(sospecden(x), type="l")
```
Index

*Topic **Density estimate**
  - bwadap.numeric, 2
  - bwplot, 5
  - bwplot.numeric, 6
*Topic **Density estimation**
  - bwadap, 2
*Topic **Flat-top Kernel**
  - bwadap.ts, 4
*Topic **Flat-top kernel**
  - bwadap.numeric, 2
  - bwplot, 5
  - bwplot.numeric, 6
  - bwplot.ts, 7
  - iospecden, 11
  - sospecden, 12
*Topic **Kernel**
  - ioksmooth, 10
*Topic **Nadaraya-Watson kernel smoother**
  - bwplot, 5
  - bwplot.numeric, 6
*Topic **Nonparametric Regression**
  - ioksmooth, 10
*Topic **Plug-in bandwidth**
  - sospecden, 12
*Topic **Smoothing parameter**
  - bwplot.ts, 7
*Topic **Spectral Density**
  - bwadap.ts, 4
  - bwplot.ts, 7
  - iospecden, 11
  - sospecden, 12
*Topic **Spectral density estimate**
  - bwplot, 5
*Topic **Spectral density estimation**
  - bwadap, 2
*Topic **density**
  - iodensity, 8
*Topic **flat-top**
  - iodensity, 8
  - bwadap, 2, 3–5, 7–9
  - bwadap.numeric, 2, 2, 9
  - bwadap.ts, 2, 3, 4, 7, 8, 11–13
  - bwplot, 3, 4, 5, 7, 8, 10, 11
  - bwplot.numeric, 3, 5, 6, 6, 7, 8, 11
  - bwplot.ts, 4, 5, 7, 12, 13
  - iodensity, 8
  - ioksmooth, 10
  - iospecden, 11
  - sospecden, 12

14