

Package ‘iosmooth’

July 22, 2025

Type Package

Title Functions for Smoothing with Infinite Order Flat-Top Kernels

Version 0.94

Date 2017-01-28

Author Timothy L. McMurry, Dimitris N. Politis

Maintainer Timothy L. McMurry <tmcmurphy@virginia.edu>

Description Density, spectral density, and regression estimation using infinite order flat-top kernels.

License GPL

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-01-28 19:16:50

Contents

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	<i>Adaptive bandwidth choice</i>
--------	----------------------------------

Description

Generic function for adaptive bandwidth choice.

Usage

```
bwadap(x, ...)
```

Arguments

x	If x is a univariate time series with class <code>ts</code> , <code>bwadap</code> returns an appropriate smoothing parameter for spectral density estimation. If x is a univariate data vector with class <code>numeric</code> , <code>bwadap</code> returns a bandwidth for density estimation.
...	Further arguments passed to the methods.

Author(s)

Timothy L. McMurry

References

Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.

See Also

[bwadap.numeric](#), [bwadap.ts](#)

bwadap.numeric	<i>Adaptive bandwidth choice for infinite order density estimates</i>
----------------	---

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, ...)
```

Arguments

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

References

Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.

See Also

[bwadap](#), [bwadap.ts](#), [bwplot.numeric](#), [bwplot](#)

Examples

```
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")
```

bwadap.ts

Adaptive bandwidth choice for infinite order spectral density estimates

Description

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

Usage

```
## 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 $c.thresh \cdot \sqrt{\log(n, 10)/n}$ and stays below that level for Kn lags.
...	Currently unimplemented.

Details

Returns a smoothing parameter l ; all lags after lag l 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 l .

Author(s)

Timothy L. McMurry

References

Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.

See Also

[bwadap](#), [bwadap](#), [bwplot](#), [bwplot.ts](#)

Examples

```
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

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

Arguments

<code>x</code>	If <code>x</code> is a univariate time series with class <code>ts</code> , <code>bwadap</code> returns an appropriate smoothing parameter for spectral density estimation. If <code>x</code> is a univariate data vector with class <code>numeric</code> , <code>bwadap</code> returns a bandwidth for density estimation.
<code>...</code>	Further arguments passed to the methods.

Author(s)

Timothy L. McMurtry

References

Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.

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

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

Arguments

<code>x</code>	A univariate data set.
<code>y</code>	A vector of responses, for regression only.
<code>smax</code>	The algorithm searches for smoothing parameters on the interval $[0, \text{smax}]$. <code>smax</code> is problem dependent, and the defaults are not consistently appropriate.
<code>normalize</code>	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 <code>FALSE</code> .
<code>n.points</code>	The number of points in $[0, \text{smax}]$ at which the sample characteristic function is calculated. If the sample characteristic function is highly oscillatory on $[0, \text{smax}]$, this may need to be increased.
<code>c.thresh</code>	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 $\text{c.thresh} \times \sqrt{\log(n, 10)/n}$ and stays below that level for a distance of Kn .
<code>...</code>	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 unperturbed 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

- Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.
- McMurry, T. L. and Politis, D. N. Minimally biased nonparametric regression and autoregression. *RevStat - Statistical Journal*, 6(2):123-150, 2008

See Also

[bwadapt](#), [bwadapt.ts](#), [bwplot.numeric](#), [bwplot](#)

Examples

```
### 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

```
## 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 $c.thresh * \sqrt{\log(n, 10)/n}$ and stays below that level for a distance of Kn .
...	Further arguments passed on to <code>plot.acf</code> .

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 unperturbed while damping out smaller higher lag correlations.

Author(s)

Timothy L. McMurry

References

Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.

See Also

[bwadapt](#), [bwadapt.ts](#), [bwplot](#), [bwplot.numeric](#)

Examples

```
set.seed(123)
x <- arima.sim(list(ar=.7, ma=-.3), 100)
bwplot(x)
bwadapt(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

```
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 bwadapt .

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; its inverse Fourier transform is estimated 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 density is estimated (x.points if specified).
y	The estimated density at the associated x values.

Author(s)

Timothy L. McMurry

References

Politis, D. N. (2001). On nonparametric function estimation with infinite-order flat-top kernels, in *Probability and Statistical Models with applications*, Ch. Charalambides et al. (Eds.), Chapman and Hall/CRC, Boca Raton, 469-483.

McMurry, T. L., & Politis, D. N. (2004). Nonparametric regression with infinite order flat-top kernels. *Journal of Nonparametric Statistics*, 16(3-4), 549-562.

See Also

[bwadap](#), [bwadap.numeric](#)

Examples

```
x <- rnorm(100)
bwplot(x)
h <- bwadap(x)
plot(iodensity(x, bw=h, kernel="Trap", n.points=300), type="l")
rug(x)
```

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

References

- McMurry, T. L., & Politis, D. N. (2004). Nonparametric regression with infinite order flat-top kernels. *Journal of Nonparametric Statistics*, 16(3-4), 549-562.
- McMurry, T. L., & Politis, D. N. (2008). Minimally biased nonparametric regression and autoregression. *REVSTAT-Statistical Journal*, 6(2), 123-150.

See Also

[bwplot](#), [bwplot.numeric](#)

Examples

```
set.seed(123)
x <- sort(runif(200, 0, 2*pi))
# Regression function
r <- exp(-x^2) + sin(x)
# Observed response
y <- r + 0.3*rnorm(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")
```

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 bwadapt.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. |

Value

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

<code>x</code>	The <code>x</code> values at which the spectral density is estimated (<code>x.points</code> if specified).
<code>y</code>	The estimated spectral density function at the associated <code>x</code> values.

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

Author(s)

Timothy L. McMurry

References

Politis, D. N., & Romano, J. P. (1995). Bias-corrected nonparametric spectral estimation. *Journal of Time Series Analysis*, 16(1), 67-103.

Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.

See Also

[bwadap.ts](#), [bwplot.ts](#)

Examples

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

sospecden

Second order spectral density estimation using an infinite order pilot

Description

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

Usage

```
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 bwadapt.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. McMurtry

References

- Politis, D. N., & Romano, J. P. (1995). Bias-corrected nonparametric spectral estimation . *Journal of Time Series Analysis*, 16(1), 67-103.
- Politis, D. N. (2003). Adaptive bandwidth choice. *Journal of Nonparametric Statistics*, 15(4-5), 517-533.

See Also

[bwadapt.ts](#), [bwplot.ts](#)

Examples

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

Index

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