Package ‘sapa’

August 29, 2016

Title Spectral Analysis for Physical Applications
Version 2.0-2
Depends R (>= 3.0.2)
License GPL-2
Imports ifultools (>= 2.0-0), splus2R (>= 1.2-0), methods
NeedsCompilation yes
Author William Constantine [cre, aut], Donald Percival [aut]
Maintainer William Constantine <wlbconstan@gmail.com>
Repository CRAN
Date/Publication 2016-05-21 00:09:16

R topics documented:

ACVS .......................... 1
SDF ........................................ 2
taper ...................................... 6

Index 11

<table>
<thead>
<tr>
<th>ACVS</th>
<th>Autocovariance sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Calculates the autocovariance sequence for an input time series.
Usage

ACVS(x, biased=TRUE, center=TRUE)

Arguments

x
  a numeric vector representing a uniformly sampled real-valued time series.

biased
  a logical value. If TRUE, the biased estimator (normalized by $N$, the number
  of samples in the time series) is returned. If FALSE, the result is the unbiased
  estimator (the kth ACVS value is normalized by $N - |k|$ for the unbiased case
  where $k = 0, \ldots, N - 1$). Default: TRUE.

center
  a logical value. If TRUE, the series is first centered (sample mean is subtracted
  from series) prior to calculating the ACVS. Default: TRUE.

Value

a numeric vector containing the single-sided ACVS for lags $k = 0, \ldots, N - 1$ where $N$ is the length
of the input time series.

See Also

SDF.

Examples

### calculate the ACVS for an N(0,1) realization
plot(seq(0,99), ACVS(rnorm(100)), type="l", lwd=2,
     xlab="lag",ylab="ACVS(rnorm(100))")
ifultools::gridOverlay()

SDF Nonparametric (cross) spectral density function estimation

Description

Estimate the process (cross) spectral density function via nonparametric models.

Usage

SDF(x, method="direct", taper=NULL, window=NULL,
     n.taper=5, overlap=0.5, blocksize=NULL,
     single.sided=TRUE, sampling.interval=NULL,
     center=TRUE, recenter=FALSE, npad=2*numRows(x))
Arguments

x a vector or matrix containing uniformly-sampled real-valued time series. If a matrix, each column should contain a different time series.

blocksize an integer representing the number of points (width) of each block in the WOSA estimator scheme. Default: floor(N/4) where N is the number of samples in each series.

center a logical value. If TRUE, the mean of each time series is recentered prior to estimating the SDF. Default: TRUE.

method a character string denoting the method to use in estimating the SDF. Choices are "direct", "lag window", "wosa" (Welch’s Overlapped Segment Averaging), "multitaper". See DETAILS for more information. Default: "direct".

n.taper an integer defining the number of tapers to use in a multitaper scheme. This value is overwritten if the taper input is of class taper. Default: 5.

npad an integer representing the total length of each time series to analyze after padding with zeros. This argument allows the user to control the spectral resolution of the SDF estimates: the normalized frequency interval is $\Delta f = 1/\text{npad}$. This argument must be set such that npad > 2. Default: $2*\text{numRows}(x)$.

overlap a numeric value on [0, 1] denoting the fraction of window overlap for the WOSA estimator. Default: 0.5.

recenter a logical value. If TRUE, the mean of each time series is recentered after (possibly) tapering the series prior to estimating the SDF. Default: FALSE.

sampling.interval a numeric value representing the interval between samples in the input time series x. Default: NULL, which serves as a flag to obtain the sampling interval via the deltaf function. If x is a list, the default sampling interval is deltaf(x[[1]]). If x is an atomic vector (ala isVectorAtomic), then the default sampling interval is established ala deltaf(x). Finally, if the input series is a matrix, the sampling interval of the first series (assumed to be in the first column) is obtained ala deltaf(x[, 1]).

single.sided a logical value. If TRUE, a single-sided SDF estimate is returned corresponding to the normalized frequency range of [0, 1/2]. Otherwise, a double-sided SDF estimate corresponding to the normalized frequency interval $[-1/2, 1/2]$ is returned. Default: TRUE.

taper. an object of class taper or a character string denoting the primary taper. If an object of class taper, the length of the taper is checked to ensure compatibility with the input x. See DETAILS for more information. The default values are a function of the method as follows:

- **direct** normalized rectangular taper
- **lag window** normalized Parzen window with a cutoff at $N/2$ where $N$ is the length of the time series.
- **wosa** normalized Hanning taper
- **multitaper** normalized Hanning taper

window an object of class taper or a character string denoting the (secondary) window for the lag window estimator. If an object of class taper, the length of the taper
is checked to ensure compatibility with the input \( x \). See DETAILS for more information. Default: Normalized Hanning window.

**Details**

Let \( X_t \) be a uniformly sampled real-valued time series of length \( N \). Let an estimate of the process spectral density function be denoted as \( \hat{S}_X(f) \) where \( f \) are frequencies on the interval \([-1/(2\Delta t), 1/(2\Delta t)]\) where \( \Delta t \) is the sampling interval. The supported SDF estimators are:

**direct** The direct SDF estimator is defined as \( \hat{S}^{(d)}_X(f) = |\sum_{t=0}^{N-1} h_t X_{t+e} e^{-i2\pi ft}|^2 \), where \( \{h_t\} \) is a data taper normalized such that \( \sum_{t=0}^{N-1} h_t^2 = 1 \). If \( h_t = 1/\sqrt{N} \) then we obtain the definition of the periodogram \( \hat{S}^{(p)}_X(f) = \frac{1}{N} |\sum_{t=0}^{N-1} X_t e^{-i2\pi ft}|^2 \). See the taper function for more details on supported window types.

**lag window** The lag window SDF estimator is defined as \( \hat{S}^{(lw)}_X(f) = \sum_{\tau=-(N-1)}^{N-1} w_\tau \hat{S}^{(d)}_{X,\tau} e^{-i2\pi f\tau} \), where \( \hat{S}^{(d)}_{X,\tau} \) is the autocovariance sequence estimator corresponding to some direct spectral estimator (often the periodogram) and \( w_\tau \) is a lag window (popular choices are the Parzen, Papoulis, and Daniell windows). See the taper function for more details.

**wosa** Welch’s Overlapped Segment Averaging SDF estimator is defined as

\[
\hat{S}^{(wosa)}_X(f) = \frac{1}{N_B} \sum_{j=0}^{N_B-1} \hat{S}^{(d)}_{j,\Delta_0}(f)
\]

where

\[
\hat{S}^{(d)}_{l}(f) \equiv \left| \sum_{t=0}^{N_S-1} h_t X_{t+l} e^{-i2\pi ft} \right|^2 , \quad 0 \leq l \leq N - N_S;
\]

Here, \( N_\Delta \) is a positive integer that controls how much overlap there is between segments and that must satisfy both \( N_\Delta \leq N_S \) and \( N_\Delta (N_B - 1) = N - N_S \), while \( \{h_t\} \) is a data taper appropriate for a series of length \( N_S \) (i.e., \( \sum_{t=0}^{N_S-1} h_t^2 = 1 \)).

**multitaper** A multitaper spectral estimator is given by

\[
\hat{S}^{(mt)}_X(f) = \frac{1}{K} \sum_{k=0}^{K-1} \left| \sum_{l=0}^{N-1} h_{k,l} X_{t+l} e^{-i2\pi ft} \right|^2 ,
\]

where \( S(k, f) = \left| \sum_{l=0}^{N-1} h_{k,l} X_{t+l} \exp(-i2\pi ft) \right|^2 \) and \( \{h_{k,l}\}, k = 0, \ldots, K - 1 \), is a set of \( K \) orthonormal data tapers.

\[
\sum_{l=0}^{N-1} h_{k,l} h_{k',l} = \begin{cases} 1, & \text{if } k = k'; \\ 0, & \text{otherwise} \end{cases}
\]

Popular choices for multitapers include sinusoidal tapers and discrete prolate spheroidal sequences (DPSS). See the taper function for more details.

**Cross spectral density function estimation:** If the input \( x \) is a matrix, where each column contains a different time series, then the results are returned in a matrix whose columns correspond to all possible unique combinations of cross-SDF estimates. For example, if \( x \) has three columns, then
the output will be a matrix whose columns are \( \{ S_{11}, S_{12}, S_{13}, S_{22}, S_{23}, S_{33} \} \) where \( S_{ij} \) is the cross-
SDF estimate of the i\(^{th} \) and j\(^{th} \) column of \( x \). All cross-spectral density function estimates are
returned as complex-valued series to maintain the phase relationships between components. For all \( S_{ij} \) where \( i = j \), however, the imaginary portions will be zero (up to a numerical noise limit).

Value

an object of class SDF.

S3 METHODS

as.matrix converts the (cross-)SDF estimate(s) as a matrix. Optional arguments are passed directly
to the matrix function during the conversion.

plot plots the (cross-)SDF estimate(s). Optional arguments are:
- xscale a character string defining the scaling to perform on the (common) frequency vec-
tor of the SDF estimates. See the scaleData function for supported choices. Default:
  "linear".
- yscale a character string defining the scaling to perform on the SDF estimates. See the
  scaleData function for supported choices. Default: "linear".
- type a single character defining the plot type (ala the par function) of the SDF plots. Default:
  ifelse(numRows(x) > 100, "l", "h").
- xlab a character string representing the x-axis label. Default: "FREQUENCY (Hz)".
- ylab a (vector of) character string(s), one per (cross-)SDF estimate, representing the y-axis
  label(s). Default: in the multivariate case, the strings "Sij" are used for the y-axis labels,
  where i and j are the indices of the different variables. For example, if the user supplies
  a 2-column matrix for \( x \), the labels "S11", "S12", and "S22" are used to label the y-axes
  of the corresponding (cross-)SDF plots. In the univariate case, the default string "SDF"
  prepended with a string describing the type of SDF performed (such as "Multitaper")
  is used to label the y-axis.
- plot.mean a logical value. If TRUE, the SDF value at normalized frequency \( f = 0 \) is plotted
  for each SDF. This frequency is associated with the sample mean of the corresponding
time series. A relatively large mean value dominates the spectral patterns in a plot and
thus the corresponding frequency is typically not plotted. Default: !attr(x,"center").
- n.plot an integer defining the maximum number of SDF plots to place onto a single graph.
  Default: 3.
- FUN a post processing function to apply to the SDF values prior to plotting. Supported
  functions are Mod, Im, Re and Arg. See each of these functions for details. If the SDF
  is purely real (no cross-SDF is calculated), this argument is coerced to the Mod function.
  Default: Mod.
- add A logical value. If TRUE, the plot is added using the current par() layout. Otherwise a
  new plot is produced. Default: FALSE.
- ... additional plot parameters passed directly to the genPlot function used to plot the SDF
  estimates.

print prints the object. Available options are:
- justify text justification ala prettPrintList. Default: "left".
- sep header separator ala prettyPrintList. Default: ":".
- ... Additional print arguments sent directly to the prettyPrintList function.
References


See Also
taper, ACVS.

Examples

```r
## calculate various SDF estimates for the
## sunspots series. remove mean component for a
## better comparison.

require(ifultools)
data <- as.numeric(sunspots)
methods <- c("direct","wosa","multitaper",
"lag window")

S <- lapply(methods, function(x, data) SDF(data, method=x), data)
x <- attr(S[[1]], "frequency")[-1]
y <- lapply(S,function(x) decibel(as.vector(x)[-1]))
names(y) <- methods

## create a stack plot of the data
stackPlot(x, y, col=1:4)

## calculate the cross-spectrum of the same
## series: all spectra should be the same in
## this case
SDF(cbind(data,data), method="lag")

## calculate the SDF using npad=31
SDF(data, npad=31, method="multitaper")
```

---

taper

Oracle function for obtaining a particular taper/window

Description

Develop signal processing tapers or windows.
taper

Usage

taper(type="rectangle", n.sample=1000, n.taper=NULL,
    sigma=0.3, beta=4*pi*(n.sample-1)/n.sample, cutoff=floor(n.sample/2),
    sidelobedB=80, roughness=n.sample/2, flatness=0.3,
    bandwidth=4, normalize=TRUE)

Arguments

bandwidth bandwidth for DPSS tapers. See Details for more information. Default: 4.

beta kaiser window shape factor (must be positive or zero). See Details for more information. Default: 4*pi*(n.sample-1)/n.sample.

cutoff parzen or Papoulis window cutoff (must be greater than unity). See Details for more information. Default: floor(n.sample/2).

flatness raised cosine taper flatness fraction (must be on [0,1]). See Details for more information. Default: 0.3.

n.sample an integer denoting the number of samples. Default: 1000.

n.taper an integer defining the multitaper order (number of orthogonal tapers) to use in a multitaper scheme. The taper order directly impacts the quality of the SDF estimate. Low taper orders are usually associated with SDF estimates with low bias and high variance, while high taper orders attenuate the variance of the estimate at the risk of incurring a large bias. This tradeoff between bias and variance is unavoidable but taper order allows you to tune the SDF to meet the needs of your application. Studies show that a multitaper order of 5 typically provides a good balance with reasonably low bias and variance properties (see the references for more details). Default: NULL, which serves as a flag to set the default taper order depending on the type of taper chosen for the analysis. If sine or dpss multitapers are chosen, the default taper order is 5, otherwise is set to unity.

normalize a logical value. If TRUE, the taper is normalized to have unit energy. Default: TRUE.

roughness daniell window down roughness factor (must be positive). See Details for more information. Default: n.sample/2.

sidelobedB chebyshev sidelobed bandwidth in decibels (must be positive). See Details for more information. Default: 80.

sigma standard deviation for Gaussian taper. Default: 0.3.

type a character string denoting the type of taper to create. Supported types are "rectangle", "triangle", "raised cosine", "hanning", "hamming", "blackman", "nuttall", "gaussian", "kaiser", "chebyshev", "born jordan", "sine", "parzen", "papoulis", "daniell", and "dpss". See Details for more information. Default: "rectangle".

Details

Let \( w(\cdot) \) and \( h(t) \) for \( t = 0, \ldots, N-1 \) be a lag window and taper, respectively. The following lag window or taper types are supported.
A rectangular taper is defined as \( h_t = 1 \).

A triangular taper is defined as \( h_t = h_{N-t-1} = 2(t+1)/(N+1) \) for \( t < M \) where \( M = \lfloor N/2 \rfloor \) and \( h_M = 1 \) if \( N \) is evenly divisible by 2.

A raised cosine is a symmetric taper with a flat mid-plateau. Let \( p \in [0,1] \) be the fraction of the length of the taper that is flat, \( M = \lfloor pN \rfloor \), and \( \beta = 2\pi/(M+1) \). A raised cosine taper is defined as

\[
h_t = h_{N-t-1} = 0.5(1 - \cos(\beta(t+1))) \quad \text{for } 0 \leq t < \lfloor M/2 \rfloor
\]

\[
h_t = 1 \quad \text{for } \lfloor M/2 \rfloor \leq t < N - \lfloor M/2 \rfloor.
\]

A Hanning taper is defined as \( h_t = 0.5(1 - \cos(\beta(t+1))) \).

A Hamming taper is defined as \( h_t = 0.54 - 0.46 \cos(\beta t) \).

A Blackman taper is defined as \( h_t = 0.42 - 0.5 \cos(\beta(t+1)) + 0.08 \cos(2\beta(t+1)) \).

A Nuttall taper is defined as \( h_t = 0.3635819 - 0.4891775 \cos(\beta t) + 0.1365995 \cos(2\beta t) - 0.0106411 \cos(3\beta t) \).

A Gaussian taper is defined as

\[
h_t = h_{N-t-1} = e^{-\beta^2/2} \quad \text{for } 0 \leq t < \lfloor N/2 \rfloor
\]

\[
h_{N/2} = 1 \quad \text{if } N \text{ is evenly divisible by 2}
\]

A Kaiser taper is defined as \( h_t = I_0(\beta \sqrt{1 - t^2}/M)/I_0(\beta) \).

The Dolph-Chebyshev taper is a function of both the desired length \( N \) and the desired sidelobe level (our routine accepts a sidelobe attenuation factor expressed in decibels). See the Mitra reference for more details.

A Born-Jordan taper is defined as \( h_t = 1/(M - t + 1) \).

Sine multitapers are defined as

\[
h_{k,t} = \left( \frac{2}{N+1} \right)^{1/2} \sin \left( \frac{(k+1)\pi(t+1)}{N+1} \right),
\]

for \( t = 0, \ldots, N-1 \) and \( k = 0, \ldots, \) . This simple equation defines a good approximation to the discrete prolate spheroidal sequences (DPSS) used in multitaper SDF estimation schemes.

A Parzen lag window is defined as

\[
w_{\tau,m} = \begin{cases} 
1 - 6(t/m)^2 + 6(|t|/m)^3, & |t| \leq m/2; \\
2(1 - |t|/m)^3, & m/2 < |t| \leq m/2; \\
0, & \text{otherwise}.
\end{cases}
\]

for \([-N+1] \leq \tau \leq (N-1)\). The variable \( m \) is referred to as the cutoff since all values beyond that point are zero.
A Papoulis lag window is defined as

\[
w_{\tau,m} = \begin{cases} \frac{1}{\pi} \sin(\pi \tau / m) + (1 - |\tau|/m) \cos(\pi \tau / m), & |\tau| < m; \\ 0, & |\tau| \geq m \end{cases}
\]

for \(-(N - 1) \leq \tau \leq (N - 1)\). The variable \(m\) is referred to as the cutoff since all values beyond that point are zero.

A Daniell lag window is defined as

\[
w_{\tau,m} = \begin{cases} \sin(\pi \tau / m), & |\tau| < N; \\ 0, & |\tau| \geq N \end{cases}
\]

for \(-(N - 1) \leq \tau \leq (N - 1)\). The variable \(m\) is referred to as the roughness factor, since, in the context of spectral density function (SDF) estimation, it controls the degree of averaging that is performed on the preliminary direct SDF estimate. The smaller the roughness, the greater the amount of smoothing.

Discrete prolate spheroidal sequences are (typically) used for multitaper spectral density function estimation. The first order DPSS can be defined (to a good approximation) as

\[
h_{t,0} = C \times I_0 \left( \tilde{W} \sqrt{1 - (1 - g_t)^2} \right) / I_0(\tilde{W})
\]

for \(t = 1, \ldots, N\), where \(C\) is a scaling constant used to force the normalization \(\sum h_{t,k}^2 = 1\); \(\tilde{W} = \pi W (N - 1) \Delta t\) where \(\Delta t\) is the sampling interval; \(g_t = (2t - 1)/N\); and \(I_0(\cdot)\) is the modified Bessel function of the first kind and zeroth order. The parameter \(W\) is related to the resolution bandwidth since it roughly defines the desired half-width of the central lobe of the resulting spectral window. Higher order DPSS tapers (i.e., \(h_{t,k}\) for \(k > 0\)) can be calculated using a relatively simple tridiagonalization formulation (see the references for more information). Finally, we note that the sampling interval \(\Delta t\) can be set to unity without any loss of generality.

Value
an object of class taper.

S3 METHODS

as.matrix converts output to a matrix.
plot plots the output. Optional arguments are:

- ylab Character string denoting the y-axis label for the plot. Default: \uppercase\(\text{attr(x,"type")}\).
- type Line type (same as the type argument of the \(\text{par}\) function). Default: "l".
- ... Additional plot arguments (set internally by the \(\text{par}\) function).
print prints a summary of the output object.

References


**See Also**

`taper`.

**Examples**

```r
require(ifultools)
## change plot layout
gap <- 0.11
old.plt <- splitplot(4,4,1,gap=gap)

## create a plot of all supported tapers and windows

for (i in seq(along=nms)){
  if (i > 1) splitplot(4,4,i,gap=gap)
  plot(taper(type=nms[i]))
}

## restore plot layout to initial state
par(old.plt)
```
Index

*Topic ts
  taper, 6

*Topic univar
  ACVS, 1
  SDF, 2

ACVS, 1, 6
as.matrix.SDF (SDF), 2
as.matrix.taper (taper), 6

plot.SDF (SDF), 2
plot.taper (taper), 6
print.SDF (SDF), 2
print.taper (taper), 6

SDF, 2, 2

taper, 6, 6, 10