# Package 'waveband'

November 4, 2024

**Title** Computes Credible Intervals for Bayesian Wavelet Shrinkage

**Version** 4.7.4

**Date** 2024-11-04

**Depends** R (>= 2.0), wavethresh (>= 4.6)

**Description** Computes Bayesian wavelet shrinkage credible intervals for
nonparametric regression.
The method uses cumulants to derive Bayesian credible intervals for
wavelet regression estimates.
The first four cumulants of the posterior distribution of the
estimates are expressed in terms of the observed data and integer
powers of the mother wavelet functions.
These powers are closely approximated by linear combinations of
wavelet scaling functions at an appropriate finer scale.
Hence, a suitable modification of the discrete wavelet transform allows
the posterior cumulants to be found efficiently for any data set.
Johnson transformations then yield the credible intervals themselves.
Barber, S., Nason, G.P. and Silverman, B.W. (2002)
<doi:10.1111/1467-9868.00332>.

**License** GPL (>= 2)

**NeedsCompilation** yes

**Author** Stuart Barber [aut],
Guy Nason [cre, ctb]

**Maintainer** Guy Nason <g.nason@imperial.ac.uk>

**Repository** CRAN

**Date/Publication** 2024-11-04 17:50:02 UTC

# Contents

**Index**                                                 **11**

---

| nmr | *Sample nmr data set* |
|-----|------------------------|

---

### Description

This page will eventually be the help page for the function you wanted.

### Usage

```
data(nmr)
```

### Format

A help page

---

| plot.wb | *Plots output from wave.band.* |
|---------|--------------------------------|

---

### Description

Plots the output from `wave.band`.

### Usage

```
## S3 method for class 'wb'
plot(x, col=FALSE, ...)
```

### Arguments

| | |
|------|----------------------------------------------------------------------|
| x | Object of class wb from the function `wave.band`. |
| col | Specifies whether the figure plotted is in colour or black and white. |
| ... | Any other arguments. |

### Details

The function `wave.band` offers a plotting option. This function will either reproduce the plot made
by `wave.band` from that function's output, or produce a different plot which reproduces better in
black and white.

## Value

A plot is produced on the current graphics device

## References

Barber, S., Nason, G.P. and Silverman, B.W. (2002) Posterior probability intervals for wavelet thresholding. *Journal of the Royal Statistical Society*, Series B, **64**, 189-206.

## See Also

[wave.band](wave.band)

---

| power.sum | *Sums of wavelets raised to integer powers* |
| --- | --- |

---

## Description

Computes a sum expressed in terms of mother wavelets raised to the power two, three, or four. Either the exact solution or a faster approximation can be computed.

## Usage

```
power.sum(alphas.wd, pow = 2, verbose = TRUE, type = "approx", plotfn = FALSE)
```

## Arguments

| | |
| --- | --- |
| alphas.wd | A wd.object, the D component of which contains the coefficients of the powers of wavelets. The entry which would normalLy be the coefficient of the wavelet at scale j and location k is the coefficient of the same wavelet raised to the power pow. |
| | If pow=2, then the overall scaling function coefficient is included in the sum, otherwise the C component is ignored completely. |
| | The filter.number and /link{family} components of alphas.wd are used to determine which wavelet is used. |
| pow | The power to which the wavelets are raised; it can take values 2, 3, or 4. |
| verbose | If verbose=TRUE, progress reports are printed while the sum is being evaluated. |
| type | If type="approx", the approximation is computed, if type="exact", the exact solution is computed, and if type="both" both the exact and approximate solutions are found. |
| plotfn | If plotfn=TRUE, the solution(s) found are plotted. |

## Details

For the approximate method, the powers of mother wavelets are represented by scaling functions (father wavelets) at a finer level. This is discussed in Barber, Nason, & Silverman (2001).

Sums of powers of wavelets are used in the computation of posterior credible intervals for wavelet regression estimators; see the documentation for the function [wave.band](wave.band) for more details.

**Value**

A vector containing the solution (either exact or approximate), or a list containing both solutions, depending on the value of "type".

**SIDE EFFECTS**

If `plotfn=TRUE`, the solution(s) found are plotted.

**References**

Barber, S., Nason, G.P. and Silverman, B.W. (2002) Posterior probability intervals for wavelet thresholding. *Journal of the Royal Statistical Society*, Series B, **64**, 189-206.

**See Also**

[wave.band](wave.band)

---

  print.wb                         *Print information about a wb object.*

---

**Description**

Print information about a wb object.

**Usage**

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

**Arguments**

x               wb object to print

...             Other arguments (not used)

**Details**

Prints information about a wb object.

**Value**

No specific value

**Author(s)**

G. P. Nason

### See Also

[plot.wb](plot.wb), [wave.band](wave.band)

### Examples

```
#
set.seed(1)
tmp <- wave.band(rnorm(32))
print(tmp)
#Wave.band credible bands object
#Bands produced for object in data component of length:  32
#Credible intervals are in the bands component
#Wave.band Bayesian hyperparameter alpha was:  0.5
#Wave.band Bayesian hyperparameter beta was:  1
#Wave.band Wavelet filter number was:  8
#Wave.band Wavelet family was:  DaubLeAsymm
#Type of input (data or test signal): data
#Rsnr (if applicable):  3
```

---

summary.wb                    *Print information about a wb object.*

---

### Description

Summarize information about a wb object.

### Usage

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

### Arguments

object          wb object to summarize

...             Other arguments (not used)

### Details

Summarizes information about a wb object.

### Value

No specific value

### Author(s)

G. P. Nason

**See Also**

    plot.wb, wave.band

**Examples**

```
#
set.seed(1)
tmp <- wave.band(rnorm(32))
summary(tmp)
#Wave.band credible bands object
#Bands produced for object in data component of length:  32
#Credible intervals are in the bands component
#Wave.band Bayesian hyperparameter alpha was:  0.5
#Wave.band Bayesian hyperparameter beta was:  1
#Wave.band Wavelet filter number was:  8
#Wave.band Wavelet family was:  DaubLeAsymm
#Type of input (data or test signal): data
#Rsnr (if applicable):  3
```

---

| test.data | *Test functions for wavelet regression and thresholding* |
|---|---|

---

**Description**

This function evaluates the "blocks", "bumps", "heavisine" and "doppler" test functions of Donoho & Johnstone (1994b) and the piecewise polynomial test function of Nason & Silverman (1994). The function also generates data sets consisting of the specified function plus uncorrelated normally distributed errors.

**Usage**

```
test.data(type = "ppoly", n = 512, signal = 1, rsnr = 7, plotfn = FALSE)
```

**Arguments**

| | |
|---|---|
| type | Test function to be computed. Available types are "ppoly" (piecewise polynomial), "blocks", "bumps", "heavi" (heavisine), and "doppler". |
| n | Number of equally spaced data points on which the function is evaluated. |
| signal | Scaling parameter; the function will be scaled so that the standard deviation of the data points takes this value. |
| rsnr | Root signal-to-noise ratio. Specifies the ratio of the standard deviation of the function to the standard deviation of the simulated errors. |
| plotfn | If plotfn=TRUE, then the test function and the simulated data set are plotted. |

**Value**

A list with the following components:

| | |
|---|---|
| x | The points at which the test function is evaluated. |
| y | The values taken by the test function. |
| ynoise | The simulated data set. |
| type | The type of function generated, identical to the input parameter type . |
| rsnr | The root signal-to-noise ratio of the simulated data set, identical to the input parameter rsnr . |

**SIDE EFFECTS**

If `plotfn=TRUE`, the test function and data set are plotted.

---

| wave.band | *Posterior credible intervals for wavelet regression* |
|---|---|

---

**Description**

Computes posterior credible intervals for an unknown regression curve.

**Usage**

```
wave.band(data = 0, alpha = 0.5, beta = 1., filter.number = 8, family =
        "DaubLeAsymm", bc = "periodic", dev = var, j0 = 3., plotfn = TRUE,
        retvalue = TRUE, n = 128, type = "data", rsnr = 3)
```

**Arguments**

Either data or a value of type other than "data" must be supplied.

| | |
|---|---|
| data | If type="data", then data should be a vector of data. The length of the vector should be a power of two not greater than 1024, and not less than 8. |
| type | Either type="data", in which case a vector of data should be supplied, or type should specify a standard test function and wave.band will generate a test data set via a call to test.data. Permissible values for type are "blocks", "bumps", "doppler", "heavi", or "ppoly"; see the documentation for test.data for more details. |
| alpha, beta | Hyperparameters which determine the priors placed on the wavelet coefficients. Both alpha and beta take positive values; see Abramovich, Sapatinas, & Silverman (1998) or Chipman & Wolfson (1999) for more details on selecting alpha and beta. |
| filter.number | A parameter relating to the smoothness of wavelet that you want to use in the decomposition. |

| family | Specifies the family of wavelets to be used. Two popular options are "DaubEx-Phase" and "DaubLeAsymm" but see the help for filter.select for more possibilities. |
| bc | Specifies the boundary handling. If bc="periodic" the default, then the function you decompose is assumed to be periodic on it's interval of definition. Other boundary options exist, but are currently unsupported for this function. |
| dev | This argument supplies the function to be used to compute the spread of the absolute values coefficients. The function supplied must return a value of spread on the variance scale (i.e. not standard deviation) such as the var() function. A popular, useful and robust alternative is the madmad function. |
| j0 | The primary resolution level; used in assessing the universal threshold which is used in the empirical Bayes estimation of hyperparameters. |
| plotfn | If plotfn=TRUE, wave.band draws the noisy data, the BayesThresh function estimate, and pointwise 99 percent credible intervals for the regression function. If the value of type is not "data", then the true function will also be plotted. |
| retvalue | If retvalue=TRUE, then a lengthy list of results will be returned. Note that if both plotfn and retvalue are set to FALSE, then wave.band will return no results whatsoever. |
| n | If type is not "data", then a data vector of length n will be generated; note that n should be a power of two not greater than 1024. |
| rsnr | If type is not "data", then the data vector generated will have root signal-to-noise ratio as specified by rsnr. |

### Details

This function implements the WaveBand method of Barber, Nason, & Silverman (2001) to compute posterior credible intervals for a regression function. The credible intervals are found by approximating the posterior distribution of the estimated regression curve at each design point. A mixture prior with two components (a zero-mean normal and a point mass at zero) is placed on each wavelet coefficient and updated by the data to give the posteriors for the wavelet coefficients. This is the same prior used by Abramovich, Sapatinas, & Silverman (1998) in their BayesThresh method, implemented in the function BAYES.THR.

The cumulants of these posteriors are computed and stored in the wd.objects returned by wave.band as Kr.wd. These are summed to give the posterior cumulants of the regression curve, which are used to fit a Johnson distribution (Johnson, 1949), using the algorithm of Hill, Hill, & Holder (1976). Percentage points of these distributions are computed by the algorithm of Hill (1976) and give the credible intervals themselves.

Code to implement the algorithms by Hill (1976) and Hill, Hill, & Holder (1976) was obtained from the StatLib archive.

### Value

If retvalue=FALSE, the value returned by wave.band is NULL. Otherwise, wave.band returns a list, an object of class wb, with the following components:

| data | The data vector which has been analysed. |

| nts | A list containing four vectors named one, two, three, and four. Vector one contains the first cumulants of the regression function estimate, vector to the second cumulants and so on. |
|---|---|
| Kr.wd | A list of four wd objects. These contain the first to fourth cumulants of the wavelet coefficients, as well as recording the wavelet used in the decomposition. |
| bands | A list containing pointwise upper and lower credible limits for the regression function estimate for nominal coverage rates 80, 90, 95 and 99 percent. The widths of the credible intervals is also included. The vectors are named with "l", "u", and "w" indicating lower limits, upper limits, and intervals widths, while "80", "90", "95", and "99" refer to the nominal coverage rate. |

The BayesThresh estimate of the regression function, using the same parameters as the WaveBand credible intervals, is also included in the pointest component of this list.

| param | A record of parameters in the call to wave.band. |
|---|---|

### SIDE EFFECTS

If plotfn=TRUE, results are plotted on the current graphics device.

### References

Barber, S., Nason, G.P. and Silverman, B.W. (2002) Posterior probability intervals for wavelet thresholding. *Journal of the Royal Statistical Society*, Series B, **64**, 189-206.

### See Also

BAYES.THR, print.wb, plot.wb, power.sum, test.data

### Examples

```
#library(wavethresh)
#
# First, look at the piecewise polynomial example.
#
# This plot and the plots for the smooth example below show
# the data as points, the BayesThresh estimate (thick line),
# pointwise 99 percent credible intervals (thin lines), and
# the true function (dotted thin line).
#
ppoly.wb <- wave.band(type = "ppoly", n = 1024, rsnr=4)
#
# Plotting the cumulants shows that there are significant
# third and fourth cumulants in some places.
#
t <- (1:1024)/1024
plot(t, ppoly.wb$cumulants$one, type="l", xlab="t", ylab = "one")
plot(t, ppoly.wb$cumulants$two, type="l", xlab="t", ylab = "two")
plot(t, ppoly.wb$cumulants$three, type="l", xlab="t", ylab = "three")
plot(t, ppoly.wb$cumulants$four, type="l", xlab="t", ylab = "four")
#
```

```
# Now consider how much difference the prior can make.
# Consider a smooth example, first using the default prior,
# and then using a smoother prior.
#

gs <- sin(2*pi*t) + 2*(t - 0.5)^2
gs.noisy <- gs + rnorm(n=1024, sd=sqrt(var(gs))/2)
gs.wb1 <- wave.band(data=gs.noisy)

gs.wb2 <- wave.band(data=gs.noisy, alpha=4, beta=1)
```

# Index