Package ‘cts’

January 14, 2019

Title  Continuous Time Autoregressive Models
Version  1.0-22
Date  2019-01-11
Author  Fortran original by G. Tunnicliffe-Wilson and Zhu Wang, R port by Zhu Wang with contribution from John Nash, Netlib and NAG authors
Description  Functions to fit continuous time autoregressive models with the Kalman filter (Wang (2013) <doi:10.18637/jss.v053.i05>).
Maintainer  Zhu Wang <wangz1@uthscsa.edu>
License  GPL (>= 2)
NeedsCompilation  yes
Repository  CRAN
Date/Publication  2019-01-14 20:10:03 UTC

R topics documented:

asth ........................................... 2
car ............................................. 2
car_control ................................... 6
factab ........................................ 8
kalsmoComp .................................. 9
plotSpecCar .................................. 10
plotSpecLs ................................... 11
spec.ci ...................................... 12
spec.ls ...................................... 12
V22174 ....................................... 14

Index  16
Measurements of The Lung Function

Description
Measurements of the lung function of an asthma patient.

Usage
data(asth)

Format
The format is: num [1:209, 1:2] 8 10 12 14 16 18 20 22 32 34 ...

Details
These are made by the patients mostly at 2h time intervals but with irregular gaps in this record.

Source
time autoregressive models for irregularly sampled time series data. *Journal of the Royal Statistical
Society, Series B, Methodological*, 56, 141–155

Examples
data(asth)
str(asth) ; plot(asth)

---

Fit Continuous Time AR Models to Irregularly Sampled Time Series

Description
Fit a continuous AR model to an irregularly sampled univariate time series with the Kalman filter

Usage
car(x, y=NULL, scale = 1.5, order = 3, ctrl=car_control())
## S3 method for class 'car'
print(x, digits = 3, ...)
## S3 method for class 'car'
summary(object, ...)
## S3 method for class 'car'
plot(x, type=c("spec", "pred", "diag"),...)

car

---

```r
## S3 method for class 'car'
predict(object, se.fit = TRUE, digits = 3, plot.it=TRUE,...)
## S3 method for class 'car'
spectrum(object, frmult=1, n.freq, plot.it = TRUE, na.action = na.fail, ...)
## S3 method for class 'car'
AIC(object, ..., k=TRUE)
## S3 method for class 'car'
tsdia(l_object, gof.lag = 10, ...)
## S3 method for class 'car'
kalsmo(object)
```

### Arguments

- **x**: two column data frame or matrix with the first column being the sampled time and the second column being the observations at the first column; otherwise `x` is a numeric vector of sampled time. It can be a `car` object for S3 methods.

- **y**: not used if `x` has two columns; otherwise `y` is a numeric vector of observations at sampled time `x`.

- **scale**: The kappa value referred to in the paper by Belcher et al. (1994). We now recommend selection of kappa along with the model order by using AIC. Also, it is suggested to choose kappa close to 2pi times reciprocal of the mean time between observations, though it is a good idea to explore somewhat lower and higher values to see whether the spectrum estimates were sensitive to this choice. Choosing kappa lower increases the risk of trying to estimate the spectrum beyond the effective Nyquist frequency of the data - though this does depend on the distribution of intersample times.

- **order**: order of autoregression.

- **ctrl**: control parameters used in `predict` and numerical optimization.

- **object**: object of class `car`.

- **type**: a character indicating the type of plot. type="spec", call spec; type="pred", call predict; type="diag", call diag.

- **se.fit**: Logical: should standard errors of prediction be returned?

- **digits**: return value digits.

- **plot.it**: Logical: plot the forecast values?

- **gof.lag**: the maximum number of lags for a Portmanteau goodness-of-fit test.

- **frmult**: numerical value, can be used to multiply the frequency range.

- **n.freq**: number of frequency.

- **k**: penalty, not used.

- **na.action**: NA action function.

- **...**: further arguments to be passed to particular methods.
Details

`spectrum` returns (and by default plots) the spectral density of the fitted model.

`tsdiag` is a generic diagnostic function for continuous AR model. It will generally plot the residuals, often standardized, the autocorrelation function of the residuals, and the p-values of a Portmanneau test for all lags up to `gof.lag`. The method for `car` object plots residuals scaled by the estimate of their (individual) variance, and use the Ljung–Box version of the portmanteau test.

`AIC` For continuous CAR model selection, t-statistic and AIC are calculated based on reparameterized coefficients `phi` and covariance matrix `ecov`. From the t-statistic, the final model is chosen such that if the true model order is less than the large value used for model estimation then for `1 > order` the deviations of the estimated parameters `phi` from their true value of 0 will be small. From the AIC, the final model is chosen based on the smallest AIC value. A table with t-statistic and AIC for the corresponding model order.

`factab` calculate characteristic roots and system frequency from the estimated reparameterized coefficients of CAR fits.

`smooth` computes components corresponding to the diagonal transition matrix with the Kalman smoother. This may not be stable for some data due to numerical inversion of matrix.

Value

A list of class "car" with the following elements:

- `n.used` The number of observations used in fitting
- `order` The order of the fitted model. This is chosen by the user.
- `np` The number of parameters estimated. This may include the mean and the observation noise ratio.
- `scale` The kappa value referred to in the paper of Belcher et al.
- `vri` If vri=1, estimate the observation noise ratio.
- `vr` The estimated observation noise ratio.
- `sigma2` The estimated innovation variance.
- `phi` The estimated reparameterized autoregressive parameters.
- `x.mean` The estimated mean of the series used in fitting and for use in prediction.
- `b` All estimated parameters, which include `phi`, and possibly `x.mean` and `vr`.
- `delb` The estimated standard error of `b`
- `essp` The estimated correlation matrix of `b`
- `ecov` The estimated covariance matrix of `phi`. See also `AIC`
- `rootr` The real part of roots of `phi`. See also `AIC`
- `rooti` The imaginary part of roots of `phi`. See also `AIC`
- `tim` The numeric vector of sampled time.
- `ser` The numeric vector of observations at sampled time `tim`.
- `filser` The filtered time series with the Kalman filter.
- `filvar` The estimated variance of Kalman filtered time series `filser`
sser  The smoothed time series with the Kalman smoother.

svar  The estimated variance of smoothed time series sser

stdred  The standardized residuals from the fitted model.

pretime  Time of predictions.

pred  Predictions for the pretime.

prv  Prediction variance of pred

pr2  Fitted values including pred for all the time series. See also fty.

prv2  Variance of fitted values including prv for all the time series. See also fty.

fty  Forecast type

tnit  Numeric vector: iteration

ss  Numeric vector: sum of squares for each tinit

bit  Matrix with rows for tinit and columns for parameter estimates

aic  AIC value for the fitted model

bic  BIC value for the fitted model

Author(s)

G. Tunnicliffe Wilson and Zhu Wang

References


See Also

car_control for predict and numerical optimization parameters, and AIC for model selection
Examples

data(V22174)
fit <- car(V22174, scale=0.2, order=7, ctrl=car_control(trace=TRUE))
summary(fit)
spectrum(fit)
tsdiag(fit)
AIC(fit)
factab(fit)
### fitted values vs observed values
ntim <- dim(V22174)[1]
plot(V22174[,1], V22174[,2], type="l")
points(V22174[,1], fit$preR[1:ntim], col="red")
### alternatively
fit2 <- car(V22174, scale=0.2, order=7, ctrl=car_control(fty=3))
plot(V22174[,1], V22174[,2], type="l")
points(V22174[,1], fit2$pre2, col="red")

data(asth)
fit <- car((asth, scale=0.25, order=4, ctrl=car_control(n.ahead=10))
kalsmo(fit)

---

car_control

Parameters for Predict and Numerical Optimization in Kalman Filter

Description
Set up in the predict and numerical optimization in the Kalman filter algorithm.

Usage

```r
car_control(fty=1, n.ahead=10, trace=FALSE, ari=TRUE, vri=FALSE, vr=0, pfi="MAPS", 
ccv="CTES", lpv=TRUE, scc=TRUE, nit=40, opm=1, rgm=1, req=0.5, con=1.0e-5, rpe=1.0, 
ivl=1.0e-2, fac=1.0e1, stl=1.0e-5, sml=1.0e2, gtl=1.0e5, kst=TRUE, fct=TRUE)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fty</code></td>
<td><code>fty=1</code> forecast past the end. <code>fty=2</code> forecast last L-steps. <code>fty=3</code> forecast last L-steps updated (filtering)types. See also <code>fct</code>. If <code>fct=TRUE</code>, all time series is used to fit the model for <code>fty=1</code> or <code>3</code>. If <code>fty=2</code>, only the first (length of time - n.ahead) is used to fit the model. Thus, only the first (length of time - n.head) prediction values are the same for <code>fty=1</code> or <code>3</code> vs <code>fty=2</code>. See also <code>pre2</code>, <code>prv2</code>.</td>
</tr>
<tr>
<td><code>n.ahead</code></td>
<td>number of steps ahead at which to predict.</td>
</tr>
<tr>
<td><code>trace</code></td>
<td>a logical value triggering printout of information during the fitting process, and major results for the fitted model.</td>
</tr>
<tr>
<td><code>ari</code></td>
<td><code>ari=TRUE</code>: parameter starting values. <code>ari=FALSE</code>: they are taken as zero. This is obsolete.</td>
</tr>
</tbody>
</table>
car_control

vri = FALSE, observation noise not included in the model. vri = TRUE, observation noise included.

vr = 0.5, initial value of observation noise ratio: only if vri = TRUE

pfi = always use the option pfi = "MAPS".

ccv = "CTES" for constant term estimation. ccv = "MNCT" if mean correction, ccv = NULL if omitted.

lpv = lpv = TRUE always use this option.

scc = scc = TRUE always use this option.

nit = number of iterations.

opm = opm = 1 always use this.

rgm = rgm = 1 always use this.

req = root equality switch value.

con = convergence criterion.

rpe = relative size of parameter perturbations.

ivl = initial value of step size constraint parameter.

fac = step size constraint modification parameter. This value may be setup to fac = 5 for better convergency.

stl = typical smallest step size parameter.

sml = typical small step size parameter.

gtl = typical greatest step size parameter.

kst = kst = TRUE to save estimated states.

fct = fct = TRUE to use all time series to fit the model.

Details

Objects returned by this function specify predict and numerical optimization parameters of the Kalman filter algorithms implemented in car, (via the ctrl argument).

Value

An object of class car_control, a list.

See Also

car for the usage
factab  
*Calculate Characteristic Roots and System Frequency*

**Description**

Calculate characteristic roots and system frequency from the estimated reparameterized coefficients of CAR fits.

**Usage**

```
factab(object)
```

**Arguments**

- `object`: a fitted time-series CAR model

**Value**

A table with characteristic roots and frequencies for the corresponding model fit.

**Author(s)**

G. Tunnicliffe Wilson and Zhu Wang

**References**


**See Also**

`car` and `kalsmo`

**Examples**

```r
data(asth)
(fit <- car(asth, scale=0.25, order=4))
factab(fit)
```
**kalsmoComp**

*Estimate Components with the Kalman Smoother*

**Description**
Estimate unobserved components with the Kalman smoother to a fitted CAR model.

**Usage**
```
kalsmoComp(x, comp = NULL, plot.it = TRUE, xlab = "time", ylab = "", na.action = na.fail, ...)```

**Arguments**
- `x`: the result of estimated components by `kalsmo`.
- `comp`: a numeric vector from which components are estimated.
- `plot.it`: plot the component?
- `xlab`: name of xlab
- `ylab`: name of ylab
- `na.action`: how to handle NAs?
- `...`: further graphical parameters.

**Value**
A component is computed from the estimated components for each root of the characteristic equation.

**Author(s)**
Zhu Wang

**References**


**See Also**
- `kalsmo`
plotSpecCar

Plotting Spectral Densities

Description

Internal function used in spectrum.car only. Not called by user. Plotting method for objects of class "spectrum.car".

Usage

plotSpecCar(x, add = FALSE, ci = 0.95, log = "dB", xlab = "frequency", ylab = NULL, type = "l", main = NULL, sub = NULL,...)

Arguments

x an object of class "spectrum.car".
add logical. If TRUE, add to already existing plot.
ci Coverage probability for confidence interval. Plotting of the confidence bar is omitted unless ci is strictly positive.
log If "dB", plot on log10 (decibel) scale (as S-PLUS), otherwise use conventional log scale or linear scale. Logical values are also accepted. The default is "yes" unless options(ts.S.compat = TRUE) has been set, when it is "dB".
xlab the x label of the plot.
ylab the y label of the plot.
type the type of plot to be drawn, defaults to lines.
main overall title for the plot.
sub a sub title for the plot.
... further graphical parameters.

Value

plot of spectral density from continuous time autoregressive model

Author(s)

G. Tunnicliffe Wilson and Zhu Wang
References

time autoregressive models for irregularly sampled time series data. Journal of the Royal Statistical
Society, Series B, Methodological, 56, 141–155

Series Analysis II, 651–682

Wang, Zhu (2004). The Application of the Kalman Filter to Nonstationary Time Series through


See Also

spectrum.car

plotSpecLs  Plotting Lomb-Scargle Periodogram

Description

Plotting method for objects of class "spec.ls".

Usage

plotSpecLs(x, add = FALSE, ci = 0.95, log = c("yes", "dB", "no"), xlab
 = "frequency", ylab = NULL, type = "l", main = NULL, sub = NULL, ...)

Arguments

x
add
.ci
.log
.xlab
.ylab
.type
.main
.sub
... an object of class "spec.car".
logical. If TRUE, add to already existing plot.
Coverage probability for confidence interval. Plotting of the confidence bar is
omitted unless .ci is strictly positive.
If "dB", plot on log10 (decibel) scale (as S-PLUS), otherwise use conventional
log scale or linear scale. Logical values are also accepted. The default is "yes"
unless options(ts.S.compat = TRUE) has been set, when it is "dB".
the x label of the plot.
the y label of the plot.
the type of plot to be drawn, defaults to lines.
overall title for the plot.
a sub title for the plot.
Further graphical parameters.

See Also

spec.ls
Description

Internal Function

Usage

spec.ls(x, y=NULL, spans = NULL, kernel = NULL, taper = 0.1, pad = 0,
fast = TRUE, type = "lomb", demean = FALSE, detrend = TRUE, plot.it = TRUE,
na.action = na.fail, ...)

Arguments

x two column data frame or matrix with the first column being the sampled time
and the second column being the observations at the first column; otherwise x is
a numeric vector of sampled time.

y not used if x has two columns; otherwise y is a numeric vector of observations
at sampled time x, the time at which x is observed

spans vector of odd integers giving the widths of modified Daniell smoothers to be
used to smooth the periodogram.

kernel alternatively, a kernel smoother of class "tskernel".

taper proportion of data to taper. A split cosine bell taper is applied to this proportion
of the data at the beginning and end of the series.

pad proportion of data to pad. Zeros are added to the end of the series to increase its
length by the proportion pad.

fast logical; if TRUE, pad the series to a highly composite length.

type Lomb-Scargle spectrum of Fourier transformation spectrum

demean logical. If TRUE, subtract the mean of the series.

detrend logical. If TRUE, remove a linear trend from the series. This will also remove the
mean.
plot.it plot the periodogram?
na.action NA action function.
... graphical arguments passed to plotSpecLs.

Details

The raw Lomb-Scargle periodogram for irregularly sampled time series is not a consistent estimator of the spectral density, but adjacent values are asymptotically independent. Hence a consistent estimator can be derived by smoothing the raw periodogram, assuming that the spectral density is smooth.

The series will be automatically padded with zeros until the series length is a highly composite number in order to help the Fast Fourier Transform. This is controlled by the fast and not the pad argument.

The periodogram at zero is in theory zero as the mean of the series is removed (but this may be affected by tapering): it is replaced by an interpolation of adjacent values during smoothing, and no value is returned for that frequency.

Value

A list object of class "spec.ls" with the following additional components:

kernel The kernel argument, or the kernel constructed from spans.
df The distribution of the spectral density estimate can be approximated by a chi square distribution with df degrees of freedom.
bandwidth The equivalent bandwidth of the kernel smoother as defined by Bloomfield (1976, page 201).
taper The value of the taper argument.
pad The value of the pad argument.
detrend The value of the detrend argument.
demean The value of the demean argument.

The result is returned invisibly if plot.it is true.

Note

This is 'slow' program and a fast program may use FFT, see (Press et al, 1992)

Author(s)

Lomb-Scargle periodogram by Zhu Wang
References


See Also

`spec.taper, plotSpecLs, fft`

Examples

```r
data(V22174)
spec.ls(V22174)

data(asth)
spec.ls(asth)
```

---

### Measurements of Relative Abundance

#### Description

Measurements of relative abundance of an oxygen isotope in an ocean core.

#### Usage

```r
data(V22174)
```

#### Format

The format is: num [1:164, 1:2] 6.13 8.39 10.64 12.90 15.16 ...

#### Source


#### References

Examples

data(V22174)
str(V22174) ; plot(V22174)
Index

*Topic datasets
  asth, 2
  V22174, 14
*Topic hplot
  plotSpecCar, 10
  plotSpecLs, 11
*Topic misc
  car_control, 6
*Topic ts
  car, 2
  factab, 8
  kalsmoComp, 9
  plotSpecCar, 10
  plotSpecLs, 11
  spec.ls, 12

AIC, 4, 5
AIC.car (car), 2
asth, 2
car, 2, 4, 7, 8
car_control, 5, 6
factab, 8
fft, 14
kalsmo, 8, 9
kalsmo (car), 2
kalsmoComp, 9
plot.car (car), 2
plotSpecCar, 10
plotSpecLs, 11, 14
predict.car (car), 2
print.car (car), 2

spec.ci, 12
spec.ls, 11, 12
spec.taper, 14
spectrum (car), 2
spectrum.car, 11
summary.car (car), 2
tsdig (car), 2
V22174, 14