Package ‘BayesSpec’

February 22, 2017

Type Package

Title Bayesian Spectral Analysis Techniques

Version 0.5.3

Author Rosen, O., Wood, S. and Stoffer, D.

Maintainer Andrew Ferris <andrew.ferris@sydney.edu.au>

Description An implementation of methods for spectral analysis using the Bayesian framework. It includes functions for modelling spectrum as well as appropriate plotting and output estimates. There is segmentation capability with RJ MCMC (Reversible Jump Markov Chain Monte Carlo). The package takes these methods predominantly from the 2012 paper ``AdaptSPEC: Adaptive Spectral Estimation for Nonstationary Time Series'' <DOI:10.1080/01621459.2012.716340>.

Imports mvtnorm (>= 1.0-5), pscl (>= 1.4.9), trust (>= 0.1-7)

License GPL-3

LazyLoad TRUE

LazyData TRUE

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-22 08:42:14

R topics documented:

adaptspec *Adaptive Spectral Estimation for Non-stationary Time Series*

Description

Methodology for analyzing possibly non-stationary time series by adaptively dividing the time series into an unknown but finite number of segments and estimating the corresponding local spectra by smoothing splines.

Usage

```
adaptspec(nloop, nwarmup, nexp_max, x,
   tmin, sigmasqalpha, tau_prior_a, tau_prior_b,
   tau_up_limit, prob_mm1, step_size_max,
   var_inflate, nbasis, nfreq_hat, plotting)
```

Arguments

nloop	The total number of MCMC iterations
nwarmup	The number of burn-in iterations
nexp_max	The maximum number of segments allowed
x	The data, a univariate time series, not a time series object
tmin	The minimum number of observations per segment. An optional argument defaulted to tmin = 40.
sigmasqalpha	An optional argument defaulted to sigmasqalpha = 100.
tau_prior_a	An optional argurment defaulted to tau_prior_a = -1.
tau_prior_b	An optional argurment defaulted to tau_prior_b = 0.
tau_up_limit	An optional argurment defaulted to tau_up_limit = 10000.
prob_mm1	An optional argurment defaulted to prob_mm1 = 0.8.
step_size_max	An optional argurment defaulted to step_size_max = 10.
var_inflate	An optional argurment defaulted to var_inflate = 1.
nbasis	An optional argurment defaulted to nbasis = 7.
nfreq_hat	An optional argurment defaulted to nfreq_hat = 50.
plotting	An optional argument for displaying output plots defaulted to FALSE. When set to TRUE, this displays the spectral and parition points.

Value

xi The partition points

log_spec_hat Estimates of the log spectra for all segments

nexp_curr The number of segments in each iteration.

Author(s)

Rosen, O., Wood, S. and Stoffer, D.

References

Rosen, O., Wood, S. and Stoffer, D. (2012). AdaptSPEC: Adaptive Spectral Estimation for Nonstationary Time Series. J. of the American Statistical Association, 107, 1575-1589

Examples

```
#Running adaptspec with the simulated_piecewise data.
data(simulated_piecewise)
model1 <- adaptspec(nloop = 80, nwarmup = 20,
   nexp_max = 5, x = simulated_piecewise[1:100])
str(model1)
summary(model1$nexp_curr)
plot(model1$nexp_curr)
```

---

intracranial_eeg *Intracranial Electroencephalograph (IEEG) Dataset*

Description

A sample of IEEG data from a subject in an interictal state.

Usage

```
data(intracranial_eeg)
```

Format

A vector time series of 6,000 observations of intracranial electroencephalograph

Source

kaggle.com

References

https://www.kaggle.com/c/melbourne-university-seizure-prediction

simulated_piecewise *Simulated Piecewise Time Series Dataset*

Description

This dataset is simulated from a piecewise autoregressive process (model (11), p. 1581, in Rosen et al. (2012)), see Examples.

Usage

```
data(simulated_piecewise)
```

Format

A univariate numeric vector with 1,024 observations.

Source

Simulated

References

Rosen, O., Wood, S. and Stoffer, D. (2012). AdaptSPEC: Adaptive Spectral Estimation for Nonstationary Time Series. J. of the American Statistical Association, 107, 1575-1589

Examples

```
#Created using the following script:
set.seed(346)
phi_true <- matrix(list(),3,1)
phi_true[[1]] <- .9
phi_true[[2]] <- c(1.69, -.81)
phi_true[[3]] <- c(1.32, -.81)
sd_true <- rep(1,3)
x1 <- arima.sim(list(order=c(1,0,0), ar=phi_true[[1]]),512,sd=sd_true[1])
x2 <- arima.sim(list(order=c(2,0,0), ar=phi_true[[2]]),256,sd=sd_true[2])
x3 <- arima.sim(list(order=c(2,0,0), ar=phi_true[[3]]),256,sd=sd_true[3])
simulated_piecewise <- c(x1, x2, x3)
plot.ts(simulated_piecewise)
```

Index