Package 'DeCAFS'

May 18, 2020

Type Package

Title Detecting Changes in Autocorrelated and Fluctuating Signals

Version 3.1.5

Date 2020-05-12

Maintainer Gaetano Romano <g.romano@lancaster.ac.uk>

Description Detect abrupt changes in time series with local fluctuations as a random walk process and autocorrelated noise as an AR(1) process. See Romano, G., Rigaill, G., Runge, V., Fearnhead, P. (2020) <arXiv:2005.01379>.

License GPL (>= 2)

Imports Rcpp (>= 1.0.0), ggplot2

LinkingTo Rcpp

NeedsCompilation yes

Depends R (>= 2.10)

LazyData true

BugReports https://github.com/gtromano/DeCAFS/issues

RoxygenNote 7.1.0

Author Gaetano Romano [aut, cre],
Guillem Rigaill [aut],
Vincent Runge [aut],
Paul Fearnhead [aut]

Repository CRAN

Date/Publication 2020-05-18 14:20:02 UTC

R topics documented:

`dataRWAR` *Generate a Random Walk + AR realization*

Description

Generate a Realization from the RWAR model (check the references for further details).

$$y_t = \mu_t + \epsilon_t$$

where

$$\mu_t = \mu_{t-1} + \eta_t + \delta_t, \quad \eta_t \sim N(0, \sigma_\eta^2),\ \delta_t\ \in R$$

and

$$\epsilon_t = \phi\epsilon_{t-1} + \nu_t \quad \nu_t \sim N(0, \sigma_\nu^2)$$

Usage

```
dataRWAR(
  n = 1000,
  poisParam = 0.01,
  meanGap = 10,
  phi = 0,
  sdEta = 0,
  sdNu = 1
)
```

Arguments

`n`	The length of the sequence of observations.
`poisParam`	A poisson parameter regulating the probability of observing a change.
`meanGap`	The average magnitude of a change.
`phi`	The autocorrelation parameter ϕ
`sdEta`	The standard deviation of the Random Walk Component on the signal drift
`sdNu`	The standard deviation of the Autocorrelated noise

Value

A list containing:

`y` the data sequence,

`signal` the underlying signal without the superimposed AR(1) noise,

`changepoints` the changepoint locations

References

Romano, G., Rigaill, G., Runge, V., Fearnhead, P. Detecting Abrupt Changes in the Presence of Local Fluctuations and Autocorrelated Noise. arXiv preprint https://arxiv.org/abs/2005.01379 (2020).

Examples

```
library(ggplot2)
set.seed(42)
Y = dataRWAR(n = 1e3, poisParam = .01, meanGap = 15, phi = .5, sdEta = 3, sdNu = 1)
y = Y$y
ggplot(data.frame(t = 1:length(y), y), aes(x = t, y = y)) +
  geom_point() +
  geom_vline(xintercept = Y$changepoints, col = 4,  lty = 3)
```

`dataSinusoidal` *Generating data from a sinusoidal model with changes*

Description

This function generates a sequence of observation from a sinusoidal model with changes. This can be used as an example for model misspecification.

Usage

```
dataSinusoidal(
  n,
  poisParam = 0.01,
  meanGap = 10,
  amplitude = 1,
  frequency = 1,
  phase = 0,
  sd = 1
)
```

Arguments

`n`	The length of the sequence of observations.
`poisParam`	A poisson parameter regulating the probability of observing a change.
`meanGap`	The average magnitude of a change.
`amplitude`	The amplitude of the sinusoid
`frequency`	The angular frequency of the sinusoid
`phase`	where the signal starts at time t = 0
`sd`	standard deviation of the noise added on top of the signal

Value

A list containing:

`y` the data sequence,

`signal` the underlying signal without the noise,

`changepoints` the changepoint locations

Examples

```
Y <- dataSinusoidal(
  1e4,
  poisParam = .0005,
  meanGap = 5,
  frequency = 2 * pi / 1e3,
  amplitude = 10,
  sd = 2
)
res <- DeCAFS(Y$y)
plot(res, col = "grey")
lines(Y$signal, col = "blue", lwd = 2, lty = 2)
abline(v = res$changepoints, col = 2)
abline(v = Y$changepoints, col = 4, lty = 2)
```

`DeCAFS` *Main DeCAFS algorithm for detecting abrupt changes*

Description

This function implements the DeCAFS algorithm to detect abrupt changes in mean of a univariate data stream in the presence of local fluctuations and auto-correlated noise. It detects the changes under a penalised likelihood model where the data, $y_1, ..., y_n$, is

$$y_t = \mu_t + \epsilon_t$$

with ϵ_t an AR(1) process, and for $t = 2, ..., N$

$$\mu_t = \mu_{t-1} + \eta_t + \delta_t$$

where at time t if we do not have a change then $\delta_t = 0$ and $\eta_t \sim N(0, \sigma_\eta^2)$; whereas if we have a change then $\delta_t \neq 0$ and $\eta_t = 0$. DeCAFS estimates the change by minimising a cost equal to twice the negative log-likelihood of this model, with a penalty β for adding a change.

Usage

```
DeCAFS(
  data,
  beta = 2 * log(length(data)),
  modelParam = estimateParameters(data),
  penalties = NULL,
  type = "std"
)
```

Arguments

`data`	A vector of observations y
`beta`	The l0 penalty. The default one is `2 * log(N)` where `N` is the length of the data.
`modelParam`	A list of 3 initial model parameters: `sdEta`, the SD of the drift (random fluctuations) in the signal, `sdNu`, the SD of the AR(1) noise process, and `phi`, the autocorrelation parameter of the noise process (so the stationary variance of the AR(1) noise process is `sdnu^2 / (1 - phi^2)`. Defaulted to `estimateParameters(data,K = 15)`, to perform automatically estimation of the three. See `estimateParameters()` for more details.
`penalties`	Can be used as an alternative to the model parameters, a list of 3 initial penalties: `lambda`, the l2-penalty penalising over the lag-1 of the signal, `gamma`, penalising over the lag-1 of the AR(1) noise process, `phi`, the autocorrelation parameter. These are related to the `modelParam` list by `list(lambda = 1 / sdEta ^ 2,gamma = 1 / sdNu ^ 2,phi = phi)`. Only one argument between `penalties` and `modelParam` should be specified. Defaulted to NULL.
`type`	The type of change one wants to look for. At the moment only 'std' is implemented.

Value

Returns an s3 object of class DeCAFSout where:

`$changepoints` is the vector of change-point locations,

`$signal` is the estimated signal without the auto-correlated noise,

`$costFunction` is the optimal cost in form of piecewise quadratics at the end of the sequence,

`$estimatedParameters` is a list of parameters estimates (if estimated, otherwise simply the initial `modelParam` input),

`$data` is the sequence of observations.

References

Romano, G., Rigaill, G., Runge, V., Fearnhead, P. Detecting Abrupt Changes in the Presence of Local Fluctuations and Autocorrelated Noise. arXiv preprint https://arxiv.org/abs/2005.01379 (2020).

Examples

```
library(ggplot2)
set.seed(42)
Y = dataRWAR(n = 1e3, poisParam = .01, meanGap = 15, phi = .5, sdEta = 1, sdNu = 3)
y = Y$y
res = DeCAFS(y)
ggplot(data.frame(t = 1:length(y), y), aes(x = t, y = y)) +
  geom_point() +
  geom_vline(xintercept = res$changepoints, color = "red") +
  geom_vline(xintercept = Y$changepoints, col = "blue",  lty = 3)
```

`estimateParameters` *Estimate parameter in the Random Walk Autoregressive model*

Description

This function perform robust estimation of parameters in the Random Walk plus Autoregressive model using a method of moments estimator. Returns a list of estimates that can be employed as an argument for parameter `modelParam` to run `DeCAFS()`.

Usage

```
estimateParameters(
  y,
  K = 15,
  phiLower = 0,
  phiUpper = 0.999,
  sdEtaUpper = Inf,
  sdNuUpper = Inf
)
```

Arguments

`y`	A vector of observations
`K`	The number of K-lags differences of the data to run the robust estimation over. Default set at 15.
`phiLower`	Smallest value of the autocorrelation parameter. Default set at 0.
`phiUpper`	Highest value of the autocorrelation parameter. Default set at 0.99.
`sdEtaUpper`	Highest value of the RW standard deviation. Default set at Inf
`sdNuUpper`	Highest value of the AR(1) noise standard deviation. Default set at Inf

Value

A list containing:

`sdEta` the SD of the drift (random fluctuations) in the signal,

`sdNu` the SD of the AR(1) noise process,

`phi` the autocorrelation parameter of the noise process.

Examples

```
set.seed(42)
y <- dataRWAR(n = 1e4, poisParam = .01, phi = .7, sdEta = 4, sdNu = 3)$y
estimateParameters(y)
```

`oilWell` *Rock structure data from an oil well*

Description

This data comes from lowering a probe into a bore-hole, and taking measurements of the rock structure as the probe is lowered. As the probe moves from one rock strata to another we expect to see an abrupt change in the signal from the measurements.

Usage

```
oilWell
```

Format

A numeric vector of 4050 obervations

Source

Ruanaidh, Joseph JK O., and William J. Fitzgerald. Numerical Bayesian methods applied to signal processing. Springer Science & Business Media, 2012. https://doi.org/10.1007/978-1-4612-0717-7

Examples

```
# removing outliers
n = length(oilWell)
h = 32
med = rep(NA, n)
for (i in 1:n) {
  index = max(1, i - h):min(n, i + h)
  med[i] = median(oilWell[index])
}
residual = (oilWell - med)

y = oilWell[abs(residual) < 8000]
sigma = sqrt(var(residual[abs(residual) < 8000]))

# running DeCAFS
res <- DeCAFS(y/sigma)
plot(res, xlab = "time", ylab = "y", type = "l")
abline(v = res$changepoints, col = 4, lty = 3)
```

`plot.DeCAFSout` *DeCAFS Plotting*

Description

DeCAFS output plotting method.

Usage

```
## S3 method for class 'DeCAFSout'
plot(x, ...)
```

Arguments

`x`	the output object from a DeCAFS call
`...`	Additional graphical parameters to be passed down to the plot function

Value

An R plot

Examples

```
set.seed(42)
Y = dataRWAR(n = 1e3, poisParam = .01, meanGap = 15, phi = .5, sdEta = 1, sdNu = 3)
res = DeCAFS(Y$y)
plot(res, type = "l")
```

Index