Package 'anomaly'

August 10, 2023

Type Package

Title Detecting Anomalies in Data

Version 4.3.0

Date 2023-08-09

Description Implements Collective And Point Anomaly (CAPA) Fisch, Eckley, and Fearnhead (2022) <doi:10.1002/sam.11586>, Multi-Variate Collective And Point Anomaly (MV-CAPA) Fisch, Eckley, and Fearnhead (2021) <doi:10.1080/10618600.2021.1987257>, Proportion Adaptive Segment Selection (PASS) Jeng, Cai, and Li (2012) <doi:10.1093/biomet/ass059>, and Bayesian Abnormal Region Detector (BARD) Bardwell and Fearnhead (2015) <arXiv:1412.5565>. These methods are for the detection of anomalies in time series data.

License GPL

Imports dplyr,tidyr,methods,assertive,ggplot2,Rcpp (>= 0.12.18),xts,zoo,Rdpack

LinkingTo Rcpp,BH

Depends R (>= 3.5.0)

NeedsCompilation yes

RoxygenNote 7.2.3

RdMacros Rdpack

Encoding UTF-8

Collate 'RcppExports.R' 'anomaly-package.R' 'generics.R' 'bard.R' 'capa.R' 'data.R' 'pass.R' 'pass.class.R'

Suggests robustbase

Author Alex Fisch [aut],
Daniel Grose [aut, cre],
Lawrence Bardwell [aut, ctb],
Idris Eckley [aut, ths],
Paul Fearnhead [aut, ths]

Maintainer Daniel Grose <dan.grose@lancaster.ac.uk>

Repository CRAN

Date/Publication 2023-08-10 07:50:09 UTC

R topics documented:

`anomaly-package` *Another package for anomoly detection*

Description

TODO - write this bit

`bard` *Detection of multivariate anomalous segments using BARD.*

Description

Implements the BARD (Bayesian Abnormal Region Detector) procedure of Bardwell and Fearnhead (2017). BARD is a fully Bayesian inference procedure which is able to give measures of uncertainty about the number and location of anomalous regions. It uses negative binomial prior distributions on the lengths of anomalous and non-anomalous regions as well as a uniform prior for the means of anomalous regions. Inference is conducted by solving a set of recursions. To reduce computational and storage costs a resampling step is included.

Usage

```
bard(
  x,
  p_N = 1/(nrow(x) + 1),
  p_A = 5/nrow(x),
  k_N = 1,
  k_A = (5 * p_A)/(1 - p_A),
  pi_N = 0.9,
  paffected = 0.05,
```

```
  lower = 2 * sqrt(log(nrow(x))/nrow(x)),
  upper = max(x),
  alpha = 1e-04,
  h = 0.25
)
```

Arguments

x	A numeric matrix with n rows and p columns containing the data which is to be inspected. The time series data classes ts, xts, and zoo are also supported.
p_N	Hyper-parameter of the negative binomial distribution for the length of non-anomalous segments (probability of success). Defaults to $\frac{1}{n+1}$.
p_A	Hyper-parameter of the negative binomial distribution for the length of anomalous segments (probability of success). Defaults to $\frac{5}{n}$.
k_N	Hyper-parameter of the negative binomial distribution for the length of non-anomalous segments (size). Defaults to 1.
k_A	Hyper-parameter of the negative binomial distribution for the length of anomalous segments (size). Defaults to $\frac{5p_A}{1-p_A}$.
pi_N	Probability that an anomalous segment is followed by a non-anomalous segment. Defaults to 0.9.
paffected	Proportion of the variates believed to be affected by any given anomalous segment. Defaults to 5%. This parameter is relatively robust to being mis-specified and is studied empirically in Section 5.1 of Bardwell and Fearnhead (2017).
lower	The lower limit of the the prior uniform distribution for the mean of an anomalous segment μ. Defaults to $2\sqrt{\frac{\log(n)}{n}}$.
upper	The upper limit of the prior uniform distribution for the mean of an anomalous segment μ. Defaults to the largest value of x.
alpha	Threshold used to control the resampling in the approximation of the posterior distribution at each time step. A sensible default is 1e-4. Decreasing alpha increases the accuracy of the posterior distribution but also increases the computational complexity of the algorithm.
h	The step size in the numerical integration used to find the marginal likelihood. The quadrature points are located from lower to upper in steps of h. Defaults to 0.25. Decreasing this parameter increases the accuracy of the calculation for the marginal likelihood but increases computational complexity.

Value

An instance of the S4 object of type .bard.class containing the data x, procedure parameter values, and the results.

Notes on default hyper-parameters

This function gives certain default hyper-parameters for the two segment length distributions. We chose these to be quite flexible for a range of problems. For non-anomalous segments a geometric distribution was selected having an average segment length of n with the standard deviation being

of the same order. For anomalous segments we chose parameters that gave an average length of 5 and a variance of n. These may not be suitable for all problems and the user is encouraged to tune these parameters.

References

Bardwell L, Fearnhead P (2017). "Bayesian Detection of Abnormal Segments in Multiple Time Series." *Bayesian Anal.*, **12**(1), 193–218.

See Also

sampler

Examples

```
library(anomaly)
data(simulated)
# run bard
bard.res<-bard(sim.data, alpha = 1e-3, h = 0.5)
sampler.res<-sampler(bard.res)
collective_anomalies(sampler.res)

plot(sampler.res,marginals=TRUE)
```

---

capa *A technique for detecting anomalous segments and points based on CAPA.*

Description

A technique for detecting anomalous segments and points based on CAPA (Collective And Point Anomalies) by Fisch et al. (2022). This is a generic method that can be used for both univariate and multivariate data. The specific method that is used for the analysis is deduced by capa from the dimensions of the data. The inputted data is either a vector (in the case of a univariate time-series) or a array with p columns (if the the time-series is p-dimensional). The CAPA procedure assumes that each component of the time-series is standardised so that the non-anomalous segments of each component have mean 0 and variance 1. This may require pre-processing/standardising. For example, using the median of each component as a robust estimate of its mean, and the mad (median absolute deviation from the median) estimator to get a robust estimate of the variance.

Usage

```
capa(
  x,
  beta,
  beta_tilde,
  type = "meanvar",
```

```
  min_seg_len = 10,
  max_seg_len = Inf,
  max_lag = 0
)
```

Arguments

x	A numeric matrix with n rows and p columns containing the data which is to be inspected. The time series data classes ts, xts, and zoo are also supported.
beta	A numeric vector of length p giving the marginal penalties. If beta is missing and p == 1 then beta = 3log(n) when the type is "mean" or "robustmean", and beta = 4log(n) otherwise. If beta is missing and p > 1, type ="meanvar" or type = "mean" and max_lag > 0 then it defaults to the penalty regime 2' described in Fisch, Eckley and Fearnhead (2022). If beta is missing and p > 1, type = "mean"/"meanvar" and max_lag = 0 it defaults to the pointwise minimum of the penalty regimes 1, 2, and 3 in Fisch, Eckley and Fearnhead (2022).
beta_tilde	A numeric constant indicating the penalty for adding an additional point anomaly. If beta_tilda is missing it defaults to 3log(np), where n and p are the data dimensions.
type	A string indicating which type of deviations from the baseline are considered. Can be "meanvar" for collective anomalies characterised by joint changes in mean and variance (the default), "mean" for collective anomalies characterised by changes in mean only, or "robustmean" (only allowed when p = 1) for collective anomalies characterised by changes in mean only which can be polluted by outliers.
min_seg_len	An integer indicating the minimum length of epidemic changes. It must be at least 2 and defaults to 10.
max_seg_len	An integer indicating the maximum length of epidemic changes. It must be at least min_seg_len and defaults to Inf.
max_lag	A non-negative integer indicating the maximum start or end lag. Only useful for multivariate data. Default value is 0.

Value

An instance of an S4 class of type capa.class.

References

Fisch ATM, Eckley IA, Fearnhead P (2022). "A linear time method for the detection of collective and point anomalies." *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **15**(4), 494-508. doi:10.1002/sam.11586.

Examples

```
library(anomaly)
# generate some multivariate data
data(simulated)
res<-capa(sim.data,type="mean",min_seg_len=2,max_lag=5)
```

```
collective_anomalies(res)
plot(res)
```

---

collective_anomalies *Collective anomaly location, lags, and mean/variance changes.*

Description

Creates a data frame containing collective anomaly locations, lags and changes in mean and variance as detected by capa, pass, and sampler.

For an object created by capa returns a data frame with columns containing the start and end position of the anomaly, the change in mean due to the anomaly. For multivariate data a data frame with columns containing the start and end position of the anomaly, the variates affected by the anomaly, as well as their the start and end lags. When type="mean"/"robustmean" only the change in mean is reported. When type="meanvar" both the change in mean and change in variance are included. If merged=FALSE (the default), then all the collective anomalies are processed individually even if they are common across multiple variates. If merged=TRUE, then the collective anomalies are grouped together across all variates that they appear in.

For an object produced by pass or sampler returns a data frame containing the start, end and strength of the collective anomalies.

Usage

```
collective_anomalies(object, ...)

## S4 method for signature 'bard.sampler.class'
collective_anomalies(object)

## S4 method for signature 'capa.class'
collective_anomalies(object, epoch = nrow(object@data), merged = FALSE)

## S4 method for signature 'pass.class'
collective_anomalies(object)
```

Arguments

object	An instance of an S4 class produced by capa.
...	TODO
epoch	Positive integer. CAPA methods are sequential and as such, can generate results up to, and including, any epoch within the data series. This can be controlled by the value of epoch and is useful for examining how the inferred anomalies are modified as the data series grows. The default value for epoch is the length of the data series.
merged	Boolean value. If merged=TRUE then collective anomalies that are common across multiple variates are merged together. This is useful when comparing the relative strength of multivariate collective anomalies. Default value is merged=FALSE. Note - merged=TRUE is currently only available when type="mean".

Value

A data frame.

See Also

capa,pass,sampler.

---

machinetemp *Machine temperature data.*

Description

Temperature sensor data of an internal component of a large, industrial machine. The data contains three known anomalies. The first anomaly is a planned shutdown of the machine. The second anomaly is difficult to detect and directly led to the third anomaly, a catastrophic failure of the machine. The data consists of 22695 observations of machine temperature recorded at 5 minute intervals along with the date and time of the measurement. The data was obtained from the Numenta Anomaly Benchmark (Ahmad et al. 2017), which can be found at https://github.com/numenta/NAB.

Usage

```
data(machinetemp)
```

Format

A dataframe with 22695 rows and 2 columns. The first column contains the date and time of the temperature measurement. The second column contains the machine temperature.

References

Ahmad S, Lavin A, Purdy S, Agha Z (2017). "Unsupervised real-time anomaly detection for streaming data." *Neurocomputing*, **262**, 134 - 147. ISSN 0925-2312, doi:10.1016/j.neucom.2017.04.070, Online Real-Time Learning Strategies for Data Streams, https://www.sciencedirect.com/science/article/pii/S0925231217309864/.

---

pass *Detection of multivariate anomalous segments using PASS.*

Description

Implements the PASS (Proportion Adaptive Segment Selection) procedure of Jeng et al. (2012). PASS uses a higher criticism statistic to pool the information about the presence or absence of a collective anomaly across the components. It uses Circular Binary Segmentation to detect multiple collective anomalies.

Usage

```
pass(x, alpha = 2, lambda = NULL, max_seg_len = 10, min_seg_len = 1)
```

Arguments

x	A numeric matrix with n rows and p columns containing the data which is to be inspected. The time series data classes ts, xts, and zoo are also supported.
alpha	A positive integer > 0. This value is used to stabilise the higher criticism based test statistic used by PASS leading to a better finite sample familywise error rate. Anomalies affecting fewer than alpha components will however in all likelihood escape detection. The default is 2.
lambda	A positive real value setting the threshold value for the familywise Type 1 error. The default value is $(1.1\log(n \times max\_seg\_len) + 2\log(\log(p)))/\sqrt{\log(\log(p))}$.
max_seg_len	A positive integer (max_seg_len > 0) corresponding to the maximum segment length. This parameter corresponds to Lmax in Jeng et al. (2012). The default value is 10.
min_seg_len	A positive integer (max_seg_len >= min_seg_len > 0) corresponding to the minimum segment length. This parameter corresponds to Lmin in Jeng et al. (2012). The default value is 1.

Value

An instance of an S4 object of type .pass.class containing the data X, procedure parameter values, and the results.

References

Jeng XJ, Cai TT, Li H (2012). "Simultaneous discovery of rare and common segment variants." *Biometrika*, **100**(1), 157-172. ISSN 0006-3444, doi:10.1093/biomet/ass059, https://academic.oup.com/biomet/article/100/1/157/193108.

Examples

```
library(anomaly)
# generate some multivariate data
data(simulated)
res<-pass(sim.data)
summary(res)
plot(res,variate_names=TRUE)
```

plot *Visualisation of data, collective and point anomalies.*

Description

Plot methods for S4 objects returned by capa, pass, and sampler.

The plot can either be a line plot or a tile plot, the type produced depending on the options provided to the plot function and/or the dimensions of the data associated with the S4 object.

Usage

```
## S4 method for signature 'bard.sampler.class'
plot(x, subset, variate_names, tile_plot, marginals = FALSE)

## S4 method for signature 'capa.class'
plot(x, subset, variate_names = FALSE, tile_plot, epoch = nrow(x@data))

## S4 method for signature 'pass.class'
plot(x, subset, variate_names = FALSE, tile_plot)
```

Arguments

x	An instance of an S4 class produced by capa, pass, or sampler.
subset	A numeric vector specifying a subset of the variates to be displayed. Default value is all of the variates present in the data.
variate_names	Logical value indicating if variate names should be displayed on the plot. This is useful when a large number of variates are being displayed as it makes the visualisation easier to interpret. Default value is FALSE.
tile_plot	Logical value. If TRUE then a tile plot of the data is produced. The data displayed in the tile plot is normalised to values in [0,1] for each variate. This type of plot is useful when the data contains are large number of variates. The default value is TRUE if the number of variates is greater than 20.
marginals	Logical value. If marginals=TRUE the plot will include visualisations of the marginal probablities of each time point being anomalous. The defualt is marginals=FALSE.
epoch	Positive integer. CAPA methods are sequential and as such, can generate results up to, and including, any epoch within the data series. This can be controlled by the value of epoch and is useful for examining how the inferred anomalies are modified as the data series grows. The default value for epoch is the length of the data series.

Value

A ggplot object.

See Also

capa,pass,sampler.

`point_anomalies` *Point anomaly location and strength.*

Description

Creates a data frame containing point anomaly locations and strengths as detected by capa.

Returns a data frame with columns containing the position, strength, and (for multivariate data) the variate number.

Usage

```
point_anomalies(object, ...)

## S4 method for signature 'capa.class'
point_anomalies(object, epoch = nrow(object@data))
```

Arguments

`object`	An instance of an S4 class produced by capa.
`...`	TODO
`epoch`	Positive integer. CAPA methods are sequential and as such, can generate results up to, and including, any epoch within the data series. This can be controlled by the value of epoch and is useful for examining how the inferred anomalies are modified as the data series grows. The default value for epoch is the length of the data series.

Value

A data frame.

See Also

capa.

`sampler` *Post processing of BARD results.*

Description

Draw samples from the posterior distribution to give the locations of anomalous segments.

Usage

```
sampler(bard_result, gamma = 1/3, num_draws = 1000)
```

Arguments

`bard_result`	An instance of the S4 class `.bard.class` containing a result returned by the `bard` function.
`gamma`	Parameter of loss function giving the cost of a false negative i.e. incorrectly allocating an anomalous point as being non-anomalous. For more details see Section 3.5 of Bardwell and Fearnhead (2017).
`num_draws`	Number of samples to draw from the posterior distribution.

Value

Returns an S4 class of type `bard.sampler.class`.

References

Bardwell L, Fearnhead P (2017). "Bayesian Detection of Abnormal Segments in Multiple Time Series." *Bayesian Anal.*, **12**(1), 193–218.

See Also

`bard`

Examples

```
library(anomaly)
data(simulated)
# run bard
res<-bard(sim.data, alpha = 1e-3, h = 0.5)
# sample
sampler(res)
```

show *Displays S4 objects produced by capa methods.*

Description

Displays S4 object produced by capa, pass, bard, and sampler. The output displayed depends on the type of S4 object passed to the method. For all types, the output indicates whether the data is univariate or multivariate, the number of observations in the data, and the type of change being detected.

Usage

```
## S4 method for signature 'bard.class'
show(object)

## S4 method for signature 'bard.sampler.class'
show(object)

## S4 method for signature 'capa.class'
show(object)

## S4 method for signature 'pass.class'
show(object)
```

Arguments

object	An instance of an S4 class produced by capa, pass, bard, or sampler.

See Also

capa,pass,bard,sampler.

sim.data *Simulated data.*

Description

A simulated data set for use in the examples and vignettes. The data consists of 500 observations on 20 variates drawn from the standard normal distribution. Within the data there are three multivariate anomalies of length 15 located at t=100, t=200, and t=300 for which the mean changes from 0 to 2. The anomalies affect variates 1 to 8, 1 to 12 and 1 to 16 respectively.

Usage

```
data(simulated)
```

Format

A matrix with 500 rows and 40 columns.

`summary` *Summary of collective and point anomalies.*

Description

Summary methods for S4 objects returned by `capa`, `pass`, and `sampler`. The output displayed depends on the type of object passed to summary. For all types, the output indicates whether the data is univariate or multivariate, the number of observations in the data, and the type of change being detected.

Usage

```
## S4 method for signature 'bard.class'
summary(object, ...)

## S4 method for signature 'bard.sampler.class'
summary(object, ...)

## S4 method for signature 'capa.class'
summary(object, epoch = nrow(object@data))

## S4 method for signature 'pass.class'
summary(object, ...)
```

Arguments

`object`	An instance of an S4 class produced by `capa` or `pass`.
`...`	Ignored.
`epoch`	Positive integer. CAPA methods are sequential and as such, can generate results up to, and including, any epoch within the data series. This can be controlled by the value of epoch and is useful for examining how the inferred anomalies are modified as the data series grows. The default value for epoch is the length of the data series.

See Also

`capa`, `pass`,`sampler`.

Index