Package 'ocd'

October 14, 2022

Type Package

Title High-Dimensional Multiscale Online Changepoint Detection

Version 1.1

Date 2020-10-02

Author Yudong Chen, Tengyao Wang, Richard J. Samworth

Maintainer Yudong Chen <yudong.chen@statslab.cam.ac.uk>

Imports stats, utils

Description

Implements the algorithm in Chen, Wang and Samworth (2020) <arxiv:2003.03668> for online detection of sudden mean changes in a sequence of high-dimensional observations. It also implements methods by Mei (2010) <doi:10.1093/biomet/asq010>, Xie and Siegmund (2013) <doi:10.1214/13-AOS1094> and Chan (2017) <doi:10.1214/17-AOS1546>.

License GPL-3

RoxygenNote 7.0.2

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-07 10:40:02 UTC

R topics documented:

`accessor` *Accessor functions to attributes of class ChangepointDetector*

Description

Accessor functions to attributes of class ChangepointDetector

Usage

```
data_dim(detector)

ocdMethod(detector)

n_obs(detector)

patience(detector)

param(detector)

thresholds(detector)

baselineMean(detector)

baselineSD(detector)

tracked(detector)

statistics(detector)

status(detector)
```

Arguments

detector	Object of S3 class 'ChangepointDetector'

Details

Obtain various attributes of the class 'ChangepointDetector'

See Also

ChangepointDetector

ChangepointDetector *Constructor for the ChangepointDetector S3 class*

Description

Constructor for the ChangepointDetector S3 class

Usage

```
ChangepointDetector(dim, method = c("ocd", "Mei", "XS", "Chan"), thresh,
  patience = 5000, MC_reps = 100, beta = 1, sparsity = "auto",
  b = beta/sqrt(dim), p0 = 1/sqrt(dim), w = 200, lambda = sqrt(8) -
  2)
```

Arguments

dim	Data dimension, all new data must be of this dimension
method	Four methods are implemented: ocd, Mei, XS and Chan. They correspond to the methods proposed in Chen, Wang and Samworth (2020), Mei (2010), Xie and Siegmund (2013) and Chan (2017). The constructed detector will be of 'OCD', 'Mei', 'XS' and 'Chan' subclass respectively.
thresh	A numeric vector or the character string 'MC'. If 'MC' is specified then the correct threshold will be computed by Monte Carlo simulation (the patience argument should be specified for this). Otherwise, for method ocd, a vector of length 3 (corresponding to the diagonal statistic, off-diagonal dense statistic and off-diagonal sparse statistic) should be specifiied; for method Mei, a vector of length two (corresponding to the max and sum statistics) should be specified; for methods XS and Chan, a single positive real number should be specified;
patience	Required patience (average run length without change) of the online change-point procedure. This is optional if the thresholds for detection are manually specified, but is required if Monte Carlo thresholds are used.
MC_reps	Number of Monte Carlo repetitions to use to estimate the thresholds. Only used when thresh='MC'.
beta	lower bound on the l_2 norm of the vector of mean change to be detected. This argument is used by the ocd method.

sparsity	Parameter used by the ocd. If sparsity='sparse', then only the diagonal and off-diagonal sparse statistics are used. If sparsity='dense', then only the diagonal and off-diagonal sparse statistics are used. If sparsity='auto', all three statistics are used to detect both sparse and dense change adaptively.
b	Lower bound on the per-coordinate magnitude of mean change be detected. This argument is used by the 'Mei' method. If b is unspecified but beta is specified, the default b = beta/sqrt(dim) will be used.
p0	A real number between 0 and 1. Sparsity parameter used by XS and Chan methods. It is the assumed fraction of nonzero coordinates of change. Default to 1/sqrt(dim).
w	Window size parameter used by XS and Chan methods. Number of most recent data points to keep track in memory. Default is 200.
lambda	A tuning parameter used by the Chan method. Default is sqrt(8)-2.

Details

This function is a wrapper. The new_OCD, new_Mei, new_XS and new_Chan carry out the actual constructor implementation.

Value

An object of S3 class 'ChangepointDetector'. Depending on the method argument specified, the object also belongs to a subclass 'OCD', 'Mei', 'XS' or 'Chan' corresponding to method='ocd'. It contains the following attributes:

- class - S3 class and subclass
- data_dim - data dimension
- method - method used for changepoint detection
- param - a list of parameters used in the specific method: beta and sparsity for method ocd; b for method Mei; p0 and w for method XS; p0, w and lambda for method Chan.
- threshold - a named vector of thresholds used for detection (see the thresh argument)
- n_obs - number of observations, initialised to 0
- baseline_mean - vector of pre-change mean, initialised to a vector of 0, can be estimated by setting the changepoint detector into baseline mean and standard deviation estimating status, see setStatus, or set directly using setBaselineMean.
- baseline_sd - vector of standard deviation, initialised to a vector of 1, can be estimated by setting the changepoint detector into baseline mean and standard deviation estimating status, see setStatus, or set directly using setBaselineSD.
- tracked - a list of information tracked online by the changepoint detector: matrices A and tail for method ocd; vector R for method Mei; matrices X_recent and CUSUM for methods XS and Chan.
- statistics - a named vector of test statistics for changepoint detection: statistics with names diag, off_d and off_s for method ocd (note if sparsity is 'dense' or 'sparse', then only (S^diag, S^off,d) and (S^diag, S^off,s) are included in stat respectively.); statistics with names max and sum for method Mei; a single numeric value for methods XS and Chan.

- status - one of the following: 'estimating' (the detector is estimating the baseline mean and standard deviation with new data points), 'monitoring' (the detector is detecting changes from the baseline mean from new data points) and an integer recording the time of declaration of changepoint.

References

- Chen, Y., Wang, T. and Samworth, R. J. (2020) High-dimensional multiscale online changepoint detection *Preprint*. arxiv:2003.03668.
- Mei, Y. (2010) Efficient scalable schemes for monitoring a large number of data streams. *Biometrika*, **97**, 419–433.
- Xie, Y. and Siegmund, D. (2013) Sequential multi-sensor change-point detection. *Ann. Statist.*, **41**, 670–692.
- Chan, H. P. (2017) Optimal sequential detection in multi-stream data. *Ann. Statist.*, **45**, 2736–2763.

See Also

accessor functions such as `data_dim`, the main function for processing a new data point `getData`, other methods for the ChangepointDetector class including `reset`, `setBaselineMean`, `setBaselineSD`, `setStatus`, `normalisedStatistics` and `checkChange`.

Examples

```
detector_ocd <- ChangepointDetector(dim=100, method='ocd',
                                    thresh=c(11.6, 179.5, 54.9), beta=1)
detector_Mei <- ChangepointDetector(dim=100, method='Mei',
                                    thresh=c(8.6, 125.1), b=0.1)
detector_XS <- ChangepointDetector(dim=100, method='XS', thresh=55.1)
detector_Chan <- ChangepointDetector(dim=100, method='Chan', thresh=8.7)
```

`Chan_update` *Processing a new data point for the 'Chan' class*

Description

This function implements the `getData` function to perform the online changepoint detection for the 'Chan' class.

Usage

```
Chan_update(x_new, X_recent, CUSUM, p0, w, lambda)
```

Arguments

x_new	a new data point
X_recent	matrix of w most recent observations
CUSUM	tail partial sums of different lengths to be tracked online
p0	sparsity parameter
w	window parameter
lambda	a tuning parameter for the 'Chan' method

Value

a list of

- stat: test statistic for the 'Chan' class.
- X_recent: the updated X_recent matrix
- CUSUM: the updated CUSUM matrix

References

Chan, H. P. (2017) Optimal sequential detection in multi-stream data. *Ann. Statist.*, **45**, 2736–2763.

---

checkChange *Check if a mean change has occurred.*

Description

Check if a mean change has occurred.

Usage

```
checkChange(detector)
```

Arguments

detector	Object of class 'Changepoint Detector'

Details

The normalisedStatistics funcrtion is used to check if any of the test statistics are above the threshold level. If this happens, the status of the detector is changed to record the time of change and a message is printed to the standard output declaring the change.

Value

Updated object detector

See Also

normalisedStatistics, setStatus,

getData *Processing a new data point*

Description

This is the main function for the 'ChangepointDetector' class.

Usage

```
getData(detector, x_new)

## S3 method for class 'OCD'
getData(detector, x_new)

## S3 method for class 'Mei'
getData(detector, x_new)

## S3 method for class 'XS'
getData(detector, x_new)

## S3 method for class 'Chan'
getData(detector, x_new)
```

Arguments

detector	Object of class 'Changepoint Detector'
x_new	A new data point. It must be of the same dimension as specified in the data_dim attribute of detector.

Details

If the status of the detector object is 'estimating', the new data point is used to update the current estimate of pre-change mean and standard deviation. If the status of the detector object is monitoring', the new data point is used to detect if a mean change has occurred.

Value

Updated object detector

Methods (by class)

- OCD: Process a new data for subclass 'OCD'
- Mei: Process a new data for subclass 'Mei'
- XS: Process a new data for subclass 'XS'
- Chan: Process a new data for subclass 'Chan'

See Also

setBaselineMean for updating the pre-change mean estimate, setBaselineSD for updating the standard deviation estimate, checkChange for checking for change.

MC_Chan *Compute Monte Carlo thresholds for the Chan method*

Description

Compute Monte Carlo thresholds for the Chan method

Usage

```
MC_Chan(dim, patience, p0, w, lambda, MC_reps)
```

Arguments

dim	Data dimension
patience	Nominal patience of the procedure
p0	Assumed fraction of nonzero coordinates of change.
w	Window size
lambda	Tuning parameter for Chan (2017) method
MC_reps	number of Monte Carlo repetitions to use

Value

A numeric vector of computed thresholds.

MC_Mei *Compute Monte Carlo thresholds for the Mei method*

Description

Compute Monte Carlo thresholds for the Mei method

Usage

```
MC_Mei(dim, patience, b, MC_reps)
```

Arguments

dim	Data dimension
patience	Nominal patience of the procedure
b	lLwer bound on per-coordinate magnitude of change
MC_reps	Number of Monte Carlo repetitions to use

Value

A numeric vector of computed thresholds.

`MC_ocd` *Compute Monte Carlo thresholds for the OCD method*

Description

Compute Monte Carlo thresholds for the OCD method

Usage

```
MC_ocd(dim, patience, beta, sparsity, MC_reps)
```

Arguments

`dim`	Data dimension
`patience`	Nominal patience of the procedure
`beta`	Lower bound on l_2 norm of change
`sparsity`	Sparsity parameter for the OCD method
`MC_reps`	Number of Monte Carlo repetitions to use

Value

A numeric vector of computed thresholds.

`MC_XS` *Compute Monte Carlo thresholds for the XS method*

Description

Compute Monte Carlo thresholds for the XS method

Usage

```
MC_XS(dim, patience, p0, w, MC_reps)
```

Arguments

`dim`	Data dimension
`patience`	Nominal patience of the procedure
`p0`	Assumed fraction of nonzero coordinates of change.
`w`	Window size
`MC_reps`	number of Monte Carlo repetitions to use

Value

A numeric vector of computed thresholds.

`Mei_update` *Processing a new data point for the 'Mei' class*

Description

This function implements the `getData` function to perform the online changepoint detection for the 'Mei' class.

Usage

```
Mei_update(x_new, R, b)
```

Arguments

`x_new`	a new data point
`R`	vector of of tail CUSUMs that will be tracked and updated online
`b`	a user specified lower bound on per-coordinate magnitude of the vector of change to be detected.

Value

a list of

- stat: a vector of 2 test statistics for the 'Mei' class.
- R: the updated R vector

References

Mei, Y. (2010) Efficient scalable schemes for monitoring a large number of data streams. *Biometrika*, **97**, 419–433.

new_Chan *construtor for subclass 'Chan' in class 'ChangepointDetector'*

Description

construtor for subclass 'Chan' in class 'ChangepointDetector'

Usage

```
new_Chan(dim, thresh, p0, w, lambda)
```

Arguments

dim	Data dimension, all new data must be of this dimension
thresh	Detection threshold. A positive real number.
p0	A sparsity parameter between 0 and 1. It is the assumed fraction of nonzero coordinates of change. Default to `1/sqrt(dim)`.
w	Window size parameter. Number of most recent data points to keep track in memory. Default is 200.
lambda	A tuning parameter used by the Chan (2017) method. Default is `sqrt(8)-2`.

Details

It is preferred to use `ChangepointDetector` for construction.

Value

An object of S3 subclass 'Chan' in class 'ChangepointDetector'.

References

Chan, H. P. (2017) Optimal sequential detection in multi-stream data. *Ann. Statist.*, **45**, 2736–2763.

Examples

```
detector <- new_Chan(dim=100, thresh=8.7, p0=0.1, w=200, lambda=sqrt(8)-2)
```

new_Mei *constructor of subclass 'Mei' in class 'ChangepointDetector'*

Description

constructor of subclass 'Mei' in class 'ChangepointDetector'

Usage

```
new_Mei(dim, thresh, b)
```

Arguments

dim	Data dimension, all new data must be of this dimension
thresh	Detection threshold. A numeric vector of length two (corresponding to the max and sum statistics) should be specified.
b	Lower bound on the per-coordinate magnitude of mean change be detected.

Details

It is preferred to use `ChangepointDetector` for construction.

Value

An object of S3 subclass 'Mei' in class 'ChangepointDetector'.

References

Mei, Y. (2010) Efficient scalable schemes for monitoring a large number of data streams. *Biometrika*, **97**, 419–433.

Examples

```
detector <- new_Mei(dim=100, thresh=c(8.6, 125.1), b=0.1)
```

new_OCD *constructor of subclass 'OCD' in class 'ChangepointDetector'*

Description

constructor of subclass 'OCD' in class 'ChangepointDetector'

Usage

```
new_OCD(dim, thresh, beta, sparsity)
```

Arguments

dim	Data dimension, all new data must be of this dimension
thresh	A numeric vector of length 3 (corresponding to the diagonal statistic, off-diagonal dense statistic and off-diagonal sparse statistic) should be specifiied.
beta	Lower bound on the l_2 norm of the vector of mean change to be detected.
sparsity	If `sparsity='sparse'`, then only the diagonal and off-diagonal sparse statistics are used. If `sparsity='dense'`, then only the diagonal and off-diagonal sparse statistics are used. If `sparsity='auto'`, all three statistics are used to detect both sparse and dense change adaptively.

Details

It is preferred to use `ChangepointDetector` for construction.

Value

An object of S3 subclass 'OCD' in class 'ChangepointDetector'.

References

Chen, Y., Wang, T. and Samworth, R. J. (2020) High-dimensional multiscale online changepoint detection *Preprint*. arxiv:2003.03668.

Examples

```
detector <- new_OCD(dim=100, thresh=c(11.6, 179.5, 54.9), beta=1, sparsity='auto')
```

---

new_XS *constructor of subclass 'XS' in class 'ChangepointDetector'*

Description

constructor of subclass 'XS' in class 'ChangepointDetector'

Usage

```
new_XS(dim, thresh, p0, w)
```

Arguments

dim	Data dimension, all new data must be of this dimension
thresh	Detection threshold. A positive real number.
p0	A sparsity parameter between 0 and 1. It is the assumed fraction of nonzero coordinates of change. Default to `1/sqrt(dim)`.
w	Window size parameter. Number of most recent data points to keep track in memory. Default is 200.

Details

It is preferred to use `ChangepointDetector` for construction.

Value

An object of S3 subclass 'XS' in class 'ChangepointDetector'.

References

Xie, Y. and Siegmund, D. (2013) Sequential multi-sensor change-point detection. *Ann. Statist.*, **41**, 670–692.

Examples

```
detector <- new_XS(dim=100, thresh=55.1, p0=0.1, w=200)
```

---

normalisedStatistics *Compute maximum ratio between detection statistic and its threshold*

Description

Compute maximum ratio between detection statistic and its threshold

Usage

```
normalisedStatistics(detector)
```

Arguments

`detector`	Object of class 'Changepoint Detector'

Value

maximum of the ratio between the current test statistics and their respective thresholds.

ocd *ocd: A package for high-dimensional multiscale online changepoint detection*

Description

The ocd package provides the S3 class `ChangepointDetector` that processes data sequentially using the `getData` function and aims to detect change as soon as it occurs online subject to false alarm rates.

References

Chen, Y., Wang, T. and Samworth, R. J. (2020) High-dimensional multiscale online changepoint detection *Preprint*. arxiv:2003.03668.

See Also

`ChangepointDetector` for detailed usage.

Examples

```
set.seed(2020)
p <- 100
thresh <- setNames(c(11.62, 179.48, 54.87), c('diag', 'off_d', 'off_s'))
detector <- ChangepointDetector(dim=p, method='ocd', beta=1, thresh=thresh)
old_mean <- rnorm(p); new_mean <- old_mean + c(rnorm(p/4), rep(0,3*p/4)) / sqrt(p/4)

# using functional semantics native in R
detector <- setStatus(detector, 'estimating')
for (i in 1:10000){
  x_new <- rnorm(p, mean=old_mean)
  detector <- getData(detector, x_new)
}
print(detector)

detector <- setStatus(detector, 'monitoring')
for (i in 1:200){
  x_new <- rnorm(p, old_mean * (i < 100) + new_mean * (i>=100))
  detector <- getData(detector, x_new)
}
print(detector)

## Not run:
# alternative way to write the above using the piping semantics
library(magrittr)
detector %<>% reset
detector %<>% setStatus('estimating')
for (i in 1:10000){
  x_new <- rnorm(p, mean=old_mean)
  detector %<>% getData(x_new)
```

```
}
detector %>% print

detector %<>% setStatus('monitoring')
for (i in 1:200){
  x_new <- rnorm(p, old_mean * (i < 100) + new_mean * (i>=100))
  detector %<>% getData(x_new)
}
detector %>% print

## End(Not run)
```

---

ocd_update *Processing a new data point for the 'OCD' class*

Description

This function implements the `getData` function to perform the online changepoint detection for the 'OCD' class.

Usage

```
ocd_update(x_new, A, tail, beta, sparsity)
```

Arguments

`x_new`	a new data point
`A`	matrix of tail CUSUMs that will be tracked and updated online
`tail`	matrix of tail lengths that will be tracked and updated online
`beta`	a user specified lower bound on the l_2 norm of the vector of change to be detected.
`sparsity`	a user specified mode parameter. If the vector of change is known to be dense or sparse, then one should set sparsity to 'dense' or 'sparse' accordingly, otherwise, the default choice sparsity='auto' will run the algorithm adaptive to the sparsity level.

Value

a list of

- stat: a vector of the test statistics for the 'OCD' class
- A: the updated A matrix
- tail: the updated tail matrix

References

Chen, Y., Wang, T. and Samworth, R. J. (2020) High-dimensional multiscale online changepoint detection *Preprint*. arxiv:2003.03668.

ParkfieldSensors *Parkfield seismic sensor data*

Description

Processed data from 39 ground motion sensors at 13 stations near Parkfield, California from 2.00-2.16am on December 23, 2004, with an earthquake measured at duration magnitude 1.47Md hit near Atascadero, California at 02:09:54.01.

Usage

```
data(ParkfieldSensors)
```

Format

A matrix with 39 columns and 14998 rows, with each column corresponding to a sensor and each row corresponding to a time after 2am. Column names corresponds to names of the sensors and row names are number of seconds after 2am.

Source

HRSN (2014), High Resolution Seismic Network. UC Berkeley Seismological Laboratory. Dataset. doi:10.7932/HRSN.

Examples

```
data(ParkfieldSensors)
head(ParkfieldSensors)

## Not run:
plot(c(0, nrow(ParkfieldSensors) * 0.064), c(0, ncol(ParkfieldSensors)+1),
     pch=' ', xlab='seconds after 2004-12-23 02:00:00',
     ylab='sensor measurements')
x <- as.numeric(rownames(ParkfieldSensors))
for (j in 1:ncol(ParkfieldSensors)){
  y <- ParkfieldSensors[, j]
  y <- (y - max(ParkfieldSensors)) / diff(range(ParkfieldSensors)) + 0.5 + j
  points(x, y, pch='.')
}
abline(v = 9 * 60 + 54.01, col='blue', lwd=2, lty=3) # earthquake time

library(magrittr)
p <- ncol(ParkfieldSensors)
train_ind <- as.numeric(rownames(ParkfieldSensors)) <= 240
train <- ParkfieldSensors[train_ind, ]
test <- ParkfieldSensors[!train_ind, ]

# tuning parameters
gamma <- 24 * 60 * 60 / 0.064 # patience = 1 day
```

```
beta <- 150

# use theoretical thresholds suggested in Chen, Wang and Samworth (2020)
psi <- function(t){p - 1 + t + sqrt(2 * (p - 1) * t)}
th_diag <- log(24*p*gamma*log2(4*p))
th_off_s <- 8*log(24*p*gamma*log2(2*p))
th_off_d <- psi(th_off_s/4)
thresh <- setNames(c(th_diag, th_off_d, th_off_s), c('diag', 'off_d', 'off_s'))

# initialise ocd detector
detector <- ChangepointDetector(dim=p, method='ocd', beta=beta, thresh=thresh)

# use training data to update baseline mean and standard deviation
detector %<>% setStatus('estimating')
for (i in 1:nrow(train)) {
  detector %<>% getData(train[i, ])
}

# find changepoint in the test data
detector %<>% setStatus('monitoring')
for (i in 1:nrow(test)) {
  detector %<>% getData(test[i, ])
  if (is.numeric(detector %>% status)) break
}

if (is.numeric(detector %>% status)) {
  time_declared <- 240 + detector %>% status * 0.064
  abline(v = time_declared, col='orange', lwd=2, lty=3) # detection time
  cat('Change detected', time_declared, 'seconds after 2am.\n')
}

## End(Not run)
```

`print.ChangepointDetector` *Printing methods for the 'ChangepointDetector' class*

Description

Printing methods for the 'ChangepointDetector' class

Usage

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

Arguments

`x`	object of the 'ChangepointDetector' class
`...`	other arguments used in `print`

`reset` *Reset changepoint detector to initial state*

Description

Reset changepoint detector to initial state

Usage

```
reset(detector)

## S3 method for class 'OCD'
reset(detector)

## S3 method for class 'Mei'
reset(detector)

## S3 method for class 'XS'
reset(detector)

## S3 method for class 'Chan'
reset(detector)
```

Arguments

`detector`	Object of class 'Changepoint Detector'

Value

Updated object `detector`

Methods (by class)

- `OCD`: Reset object of subclass 'OCD'
- `Mei`: Reset object of subclass 'Mei'
- `XS`: Reset object of subclass 'XS'
- `Chan`: Reset object of subclass 'Chan'

setBaselineMean *Set baseline mean*

Description

Set baseline mean

Usage

```
setBaselineMean(detector, mean)
```

Arguments

detector	Object of class 'Changepoint Detector'
mean	vector of pre-change mean, must be of the same dimension as specified in the data_dim attribute of detector.

Value

Updated object detector

setBaselineSD *Set baseline standard deviation*

Description

Set baseline standard deviation

Usage

```
setBaselineSD(detector, sd)
```

Arguments

detector	Object of class 'Changepoint Detector'
sd	vector of standard deviation, must be of the same dimension as specified in the data_dim attribute of detector.

Value

Updated object detector

setStatus *Set changepoint detector status*

Description

Set changepoint detector status

Usage

```
setStatus(detector, new_status)
```

Arguments

detector	Object of class 'Changepoint Detector'
new_status	'estimating' or 'monitoring'

Details

If the status is set to 'estimating', new observations are used to update current estimate of pre-change mean and standard deviation. If the status is set to 'monitoring', new observations are used to check if mean change has occurred.

Value

Updated object detector

update_param *compute new mean and sd from old ones with one additional observation*

Description

compute new mean and sd from old ones with one additional observation

Usage

```
update_param(old_mean, old_sd, x_new, n_obs)
```

Arguments

old_mean	vector of old means
old_sd	vector of old standard deviation
x_new	new observation vector
n_obs	total number of observations (including x_new)

Value

list of two vectors: new mean and new standard deviation

XS_update *Processing a new data point for the 'XS' class*

Description

This function implements the getData function to perform the online changepoint detection for the 'XS' class.

Usage

```
XS_update(x_new, X_recent, CUSUM, p0, w)
```

Arguments

x_new	a new data point
X_recent	matrix of w most recent observations
CUSUM	tail partial sums of different lengths to be tracked online
p0	sparsity parameter
w	window parameter

Value

a list of

- stat: test statistic for the 'XS' class.
- X_recent: the updated X_recent matrix
- CUSUM: the updated CUSUM matrix

References

Xie, Y. and Siegmund, D. (2013) Sequential multi-sensor change-point detection. *Ann. Statist.*, **41**, 670–692.

Index