Package 'ffstream'

November 22, 2016

Title Forgetting Factor Methods for Change Detection in Streaming Data

Version 0.1.5

Author Dean Bodenham

Maintainer Dean Bodenham <deanbodenhambsse@gmail.com>

Description An implementation of the adaptive forgetting factor scheme described in Bodenham and Adams (2016) <doi:10.1007/s11222-016-9684-8> which adaptively estimates the mean and variance of a stream in order to detect multiple changepoints in streaming data. The implementation is in C++ and uses Rcpp. Additionally, implementations of the fixed forgetting factor scheme from the same paper, as well as the classic CUSUM and EWMA methods, are included.

Depends R (>= 3.3.0)

License GPL-2 | GPL-3

LazyData true

LinkingTo Rcpp

Imports methods, Rcpp (>= 0.12.7)

Suggests testthat (>= 1.0.2), knitr, rmarkdown

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-11-22 18:23:27

R topics documented:

`computeAFFMean` *Quick computation of AFF mean of a given vector*

Description

Given a vector `x` and a value `eta` for step-size in the stochastic gradient descent for the adaptive forgetting factor, this returns the value of the fixed forgetting factor mean $\bar{x}_{N,\overrightarrow{\lambda}}$, where N is the length of `x`. Algorithm is implemented in C++.

Usage

```
computeAFFMean(x = c(0), eta = 0.01)
```

Arguments

`x`	Vector of numeric values values. Default is `c(0)`, a vector of one element (zero)
`eta`	Value for the step size in the gradient descent step. Default is `eta=0.01`.

Author

Dean Bodenham

References

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation*. Statistics and Computing doi:10.1007/s11222-016-9684-8

See Also

`computeFFFMean`

computeFFFMean *Quick computation of FFF mean of a given vector*

Description

Given a vector x and a value lambda for a fixed forgetting factor, returns the value of the fixed forgetting factor mean $\bar{x}_{N,\lambda}$, where N is the length of x. Algorithm is implemented in C++.

Usage

```
computeFFFMean(x = c(0), lambda = 0.99)
```

Arguments

x	Vector of numeric values values. Default is c(0), a vector of one element (zero)
lambda	Value for the fixed forgetting factor in $[0, 1]$. Default is lambda=0.99.

Author

Dean Bodenham

References

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation.* Statistics and Computing doi:10.1007/s11222-016-9684-8

See Also

computeAFFMean

demo_ffstream *Demo for ffstream*

Description

Provides a demonstration of the AFF method detecting changepoints in a stream.

Usage

```
demo_ffstream(showPlot = FALSE, returnStream = FALSE, plotSmall = FALSE)
```

Arguments

showPlot	Boolean flag; if TRUE, then a plot is generated. Default is FALSE.
returnStream	Boolean flag; if TRUE, then return the stream as part of the list returned by the demo. Default is FALSE.
plotSmall	Boolean flag; if TRUE, creates a small plot, as needed for the vignette. Default is FALSE.

Details

This method generates a stream with three changepoints, and finds the changepoints with AFF. Also creates a plot of the data and the changepoints if the showPlot flag is set to TRUE. The observations are shown in black, the true changepoints are shown as red dotted vertical lines, and the detected (estimated) changepoints are shown as blue dashed lines. The following is returned in a list:

tau The location of the true changepoints.

tauhat The detected (estimated) changepoints.

method The method used, in this case AFF.

param The data frame with the parameters used in the AFF method, in this case,

- **alpha** The significance level,
- **eta** The step size in the gradient descent, whose value is not particularly important,
- **BL** The length of the burn-in period.

Author

Dean Bodenham

References

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation.* Statistics and Computing doi:10.1007/s11222-016-9684-8

Examples

```
df <- demo_ffstream()

## Not run:
demo_ffstream(showPlot=TRUE)

## End(Not run)
```

detectAFFMean *Detect a change/changes in a vector using AFF method*

Description

Given a vector x, use the fFF method to sequentially detect changes (or a single change) in the MEAN of the vector.

Usage

```
detectAFFMean(x, alpha = 0.01, eta = 0.01, BL = 50, multiple = TRUE,
  single = !multiple, usePrechange = FALSE, prechangeMean = NULL,
  prechangeSigma = NULL, prechangeVar = NULL, skipCheck = FALSE)
```

Arguments

x	The vector (stream) in which to detect change(s).
alpha	The value for the threshold. Default is alpha=0.01.
eta	The value for of the step size in the gradient descent step. Results show that values of $0.1, 0.01, 0.001$ all produce similar results. Default is eta=0.01.
BL	The burn-in length. Default is BL=50.
multiple	Boolean to use to decide whether to detect multiple changes or only a single change. Default is TRUE (i.e. detect multiple changes).
single	Boolean to use to decide whether to detect only a single change or multiple changes. Set to !multiple, i.e. default is FALSE. If both single and multiple are set to TRUE, then only a single change will be detected; if both set to FALSE then multiple changes will be detected (i.e. single dominates).
usePrechange	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is FALSE. If TRUE, then prechange mean and standard deviation variance must be specified. See parameters prechangeMean, prechangeSigma and prechangeVar.
prechangeMean	Value to be used for the prechange mean. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error.
prechangeSigma	Value to be used for the prechange standard deviation. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeVar is not NULL.
prechangeVar	Value to be used for the prechange variance. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeSigma is not NULL. prechangeVar is set to sqrt(prechangeSigma).
skipCheck	A boolean which allows the function to skip the check of the stream. Default is FALSE.

Value

A list with the following elements:

`tauhat` A vector of the changepoints found.

Author

Dean Bodenham

References

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation.* Statistics and Computing doi:10.1007/s11222-016-9684-8

Examples

```
# create a stream with three changepoints
set.seed(8)
x <- rnorm(400, 5, 1) + rep(c(0:3), each=100) # mean is 5 and s.d. is 1

# multiple changepoints
list_aff <- detectAFFMean(x, alpha=0.01, eta=0.01, BL=50, multiple=TRUE)

# now only a single (the first) changepoint
list_aff2 <- detectAFFMean(x, alpha=0.01, eta=0.01, BL=50, single=TRUE)

# now only a single (the first) changepoint, but with the prechange
# mean and variance known
list_aff3 <- detectAFFMean(x, alpha=0.01, eta=0.01, single=TRUE,
                           prechangeMean=5, prechangeSigma=1)
```

`detectCUSUMMean` *Detect a change/changes in a vector using CUSUM method*

Description

Given a vector `x`, use the CUSUM method to sequentially detect changes (or a single change) in the MEAN of the vector.

Usage

```
detectCUSUMMean(x, k = 0.25, h = 8, BL = 50, multiple = TRUE,
  single = !multiple, usePrechange = FALSE, prechangeMean = NULL,
  prechangeSigma = NULL, prechangeVar = NULL, skipCheck = FALSE)
```

Arguments

x	The vector (stream) in which to detect change(s).
k	control parameter for CUSUM. Default is 0.25.
h	control parameter for CUSUM. Defqult is 8.00.
BL	The burn-in length. Default is BL=50.
multiple	Boolean to use to decide whether to detect multiple changes or only a single change. Default is TRUE (i.e. detect multiple changes).
single	Boolean to use to decide whether to detect only a single change or multiple changes. Set to !multiple, i.e. default is FALSE. If both single and multiple are set to TRUE, then only a single change will be detected; if both set to FALSE then multiple changes will be detected (i.e. single dominates).
usePrechange	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is FALSE. If TRUE, then prechange mean and standard deviation variance must be specified. See parameters prechangeMean, prechangeSigma and prechangeVar.
prechangeMean	Value to be used for the prechange mean. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error.
prechangeSigma	Value to be used for the prechange standard deviation. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeVar is not NULL.
prechangeVar	Value to be used for the prechange variance. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeSigma is not NULL. prechangeVar is set to sqrt(prechangeSigma).
skipCheck	A boolean which allows the function to skip the check of the stream. Default is FALSE.

Details

CUSUM updates via:

$$S_j = \max 0, S_{j-1} + (x_j - \mu)/\sigma - k$$

and

$$T_j = \max 0, S_{j-1} - (x_j - \mu)/\sigma - k$$

where μ and σ are, respectively, the mean and variance of the in-control stream, x_j is the observation at time j and k is a control parameter for CUSUM. Then, a change is signalled if $S_j > h$ or $T_j > h$, where h is the other control parameter. This is the formulation for using CUSUM to detect an increase or decrease in the mean.

Value

A list with the following elements:

tauhat A vector of the changepoints found.

Author

Dean Bodenham

References

E. S. Page (1954) *Continuous inspection schemes*. Biometrika, 41(1/2), 100-115

Examples

```
# create a stream with three changepoints
set.seed(8)
x <- rnorm(400, 5, 1) + rep(c(0:3), each=100) # mean is 5 and s.d. is 1

# multiple changepoints
list_cusum <- detectCUSUMMean(x, k=0.25, h=8.00, BL=50, multiple=TRUE)

# now only a single (the first) changepoint
list_cusum2 <- detectCUSUMMean(x, k=0.25, h=8.00, BL=50, single=TRUE)

# now only a single (the first) changepoint, but with the prechange
# mean and variance known
list_cusum3 <- detectCUSUMMean(x, k=0.25, h=8.00, BL=50, single=TRUE,
                               prechangeMean=5, prechangeSigma=1)
```

`detectEWMAMean` *Detect a change/changes in a vector using EWMA method*

Description

Given a vector `x`, use the EWMA method to sequentially detect changes (or a single change) in the MEAN of the vector.

Usage

```
detectEWMAMean(x, r = 0.25, L = 3, BL = 50, multiple = TRUE,
  single = !multiple, usePrechange = FALSE, prechangeMean = NULL,
  prechangeSigma = NULL, prechangeVar = NULL, skipCheck = FALSE)
```

Arguments

`x`	The vector (stream) in which to detect change(s).
`r`	Control parameter for EWMA. Must be in range $[0, 1]$. Default is `r=0.25`.
`L`	Control parameter for EWMA. Default is `L=3.00`
`BL`	The burn-in length. Default is `BL=50`.
`multiple`	Boolean to use to decide whether to detect multiple changes or only a single change. Default is `TRUE` (i.e. detect multiple changes).

single	Boolean to use to decide whether to detect only a single change or multiple changes. Set to !multiple, i.e. default is FALSE. If both single and multiple are set to TRUE, then only a single change will be detected; if both set to FALSE then multiple changes will be detected (i.e. single dominates).
usePrechange	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is FALSE. If TRUE, then prechange mean and standard deviation variance must be specified. See parameters prechangeMean, prechangeSigma and prechangeVar.
prechangeMean	Value to be used for the prechange mean. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error.
prechangeSigma	Value to be used for the prechange standard deviation. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeVar is not NULL.
prechangeVar	Value to be used for the prechange variance. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeSigma is not NULL. prechangeVar is set to sqrt(prechangeSigma).
skipCheck	A boolean which allows the function to skip the check of the stream. Default is FALSE.

Details

EWMA updates via:

$$Z_j = (1 - r)Z_{j-1} + rx_j$$

where μ is the mean of the in-control stream, x_j is the observation at time j and r is a control parameter for EWMA. Then, a change is signalled if

$$|Z_j - \mu| > L\sigma_{Z_j}$$

, where L is the other control parameter, and σ_{Z_j} is a scaled version of the in-control variance σ. This is the formulation for using EWMA to detect an increase or decrease in the mean.

Value

A list with the following elements:

tauhat A vector of the changepoints found.

Author

Dean Bodenham

References

S. W. Roberts (1959) *Control chart tests based on geometric moving averages*. Technometrics, 1(3), 239-250

Examples

```
# create a stream with three changepoints
set.seed(8)
x <- rnorm(400, 5, 1) + rep(c(0:3), each=100) # mean is 5 and s.d. is 1

# multiple changepoints
list_ewma <- detectEWMAMean(x, r=0.25, L=3.023, BL=50, multiple=TRUE)

# now only a single (the first) changepoint
list_ewma2 <- detectEWMAMean(x, r=0.25, L=3.023, BL=50, single=TRUE)

# now only a single (the first) changepoint, but with the prechange
# mean and variance known
list_ewma3 <- detectEWMAMean(x, r=0.25, L=3.023, BL=50, single=TRUE,
                             prechangeMean=5, prechangeSigma=1)
```

---

detectFFFMean *Detect a change/changes in a vector using FFF method*

Description

Given a vector `x`, use the FFF method to sequentially detect changes (or a single change) in the MEAN of the vector.

Usage

```
detectFFFMean(x, lambda = 0.95, alpha = 0.01, BL = 50, multiple = TRUE,
  single = !multiple, usePrechange = FALSE, prechangeMean = NULL,
  prechangeSigma = NULL, prechangeVar = NULL, skipCheck = FALSE)
```

Arguments

`x`	The vector (stream) in which to detect change(s).
`lambda`	The value for the forgetting factor. Default is `lambda=0.95`.
`alpha`	The value for the threshold. Default is `alpha=0.01`.
`BL`	The burn-in length. Default is `BL=50`.
`multiple`	Boolean to use to decide whether to detect multiple changes or only a single change. Default is `TRUE` (i.e. detect multiple changes).
`single`	Boolean to use to decide whether to detect only a single change or multiple changes. Set to `!multiple`, i.e. default is FALSE. If both `single` and `multiple` are set to `TRUE`, then only a single change will be detected; if both set to `FALSE` then multiple changes will be detected (i.e. `single` dominates).

usePrechange	Boolean indicating whether prechange parameters (mean and variance) are known and will be used (or not). Default is FALSE. If TRUE, then prechange mean and standard deviation variance must be specified. See parameters prechangeMean, prechangeSigma and prechangeVar.
prechangeMean	Value to be used for the prechange mean. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error.
prechangeSigma	Value to be used for the prechange standard deviation. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeVar is not NULL.
prechangeVar	Value to be used for the prechange variance. Default is NULL. If prechangeKnown = TRUE and value is NULL, this will result in an error, unless prechangeSigma is not NULL. prechangeVar is set to sqrt(prechangeSigma).
skipCheck	A boolean which allows the function to skip the check of the stream. Default is FALSE.

Value

A list with the following elements:

tauhat A vector of the changepoints found.

Author

Dean Bodenham

References

D. A. Bodenham and N. M. Adams (2016) *Continuous monitoring for changepoints in data streams using adaptive estimation*. Statistics and Computing doi:10.1007/s11222-016-9684-8

Examples

```
# create a stream with three changepoints
set.seed(8)
x <- rnorm(400, 5, 1) + rep(c(0:3), each=100) # mean is 5 and s.d. is 1

# multiple changepoints
list_fff <- detectFFFMean(x, alpha=0.01, lambda=0.95, BL=50, multiple=TRUE)

# now only a single (the first) changepoint
list_fff2 <- detectFFFMean(x, alpha=0.01, lambda=0.95, BL=50, single=TRUE)

# now only a single (the first) changepoint, but with the prechange
# mean and variance known
list_fff3 <- detectFFFMean(x, alpha=0.01, lambda=0.95, BL=50, single=TRUE,
                           prechangeMean=5, prechangeSigma=1)
```

initAFFMean *Initialisation of AFF mean*

Description

This function makes it simple to initalise an AFF object.

Usage

```
initAFFMean(eta = 0.1)
```

Arguments

eta	The value of the step size in the gradient descent. Default is eta=0.01.

Examples

```
library(Rcpp)
aff1 <- initAFFMean()     # initialises with eta=0.01

aff2 <- initAFFMean(0.1)  # initialises with eta=0.1
```

initAFFMeanCD *Initialisation of AFF change detector*

Description

This function makes it simple to initalise an FFF object.

Usage

```
initAFFMeanCD(alpha, eta = 0.01, BL = 50)
```

Arguments

alpha	The value of the significance level. There is no default value on purpose - it must be specified by the user.
eta	The value of the step-size in the gradient descent. Default is eta=0.01.
BL	The length of the burn-in region. Default value is BL=50.

Examples

```
library(Rcpp)
affmeancd1 <- initAFFMeanCD(alpha=0.01)    # initialises with eta=0.01,
                                           # and BL=50

affmeancd2 <- initAFFMeanCD(alpha=0.005, eta=0.1,  BL=100)
```

---

initCUSUMMeanCD *Initialisation of CUSUM*

Description

This function makes it simple to initalise a CUSUM object.

Usage

```
initCUSUMMeanCD(k = 0.25, h = 8, BL = 50)
```

Arguments

k	One of the CUSUM control parameters. Default value is k=0.25.
h	One of the CUSUM control parameters. Default value is h=8.00.
BL	The burn-in length to be used with a CUSUM change detector. Default value is BL=50.

Examples

```
library(Rcpp)
c1 <- initCUSUMMeanCD()     # initialises with k=0.25, h=8.00, BL=50

c2 <- initCUSUMMeanCD(k=0.5, h=4.00, BL=30)
```

---

initEWMAMeanCD *Initialisation of EWMA*

Description

This function makes it simple to initialise a EWMA object.

Usage

```
initEWMAMeanCD(r = 0.2, L = 3, BL = 50)
```

Arguments

r	One of the EWMA control parameters. Default value is r=0.20.
L	One of the EWMA control parameters. Default value is L=3.00.
BL	The burn-in length to be used with a EWMA change detector. Default value is BL=50.
	library(Rcpp) e1 <- initEWMAMeanCD() #initialises with r=0.20, L=3.00
	e1 <- initEWMAMeanCD(r=0.05, L=0.275) #initialises with r=0.20, L=3.00

---

initFFFMean *Initialisation of FFF*

Description

This function makes it simple to initalise an FFF mean object.

Usage

```
initFFFMean(lambda = 1)
```

Arguments

lambda	The value of the fixed forgetting factor. Default is lambda=1.

Examples

```
library(Rcpp)
fff1 <- initFFFMean()     # initialises with lambda=1

fff2 <- initFFFMean(0.9)  # initialises with lambda=0.9
```

---

initFFFMeanCD *Initialisation of FFF mean change detector*

Description

This function makes it simple to initalise an FFF object.

Usage

```
initFFFMeanCD(alpha, lambda = 1, BL = 50)
```

Arguments

alpha	The value of the significance level. There is no default value on purpose - it must be set by the user.
lambda	The value of the fixed forgetting factor. Default value is lambda=1.
BL	The length of the burn-in region. Default value is BL=50.

Examples

```
library(Rcpp)
fffcd1 <- initFFFMeanCD(0.99, 0.95)     # initialises with alpha=0.99
                                        # and lambda=0.95 (and BL=50)
```

Index