Package ‘ffstream’

May 14, 2018

Title Forgetting Factor Methods for Change Detection in Streaming Data
Version 0.1.6
Date 2018-05-12
Author Dean Bodenham
Maintainer Dean Bodenham <deanbodenhampkgs@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 (>= 3.5.0), Rcpp (>= 0.12.16)
License GPL-2 | GPL-3
URL http://www.deanbodenham.com/ffstream
LazyData true
LinkingTo Rcpp
Imports methods
Suggests testthat (>= 2.0.0), knitr, rmarkdown
Encoding UTF-8
RoxygenNote 6.0.1
VignetteBuilder knitr
NeedsCompilation yes
Repository CRAN
Date/Publication 2018-05-14 08:03:21 UTC

R topics documented:

computeAFFMean ............................................. 2
computeFFFMean ............................................. 3
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 \( \bar{x}_{N, \lambda} \), where \( N \) is the length of \( x \). Algorithm is implemented in C++.

Usage

\[
\text{computeAFFMean}(x = c(0), \eta = 0.01)
\]

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


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

\[
\text{computeFFFMean}(x = c(0), \lambda = 0.99)
\]

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


See Also

computeAFFMean

demo_ffstream

Demo for ffstream

Description

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

Usage

\[
\text{demo_ffstream}(\text{showPlot} = \text{FALSE}, \text{returnStream} = \text{FALSE}, \text{plotSmall} = \text{FALSE})
\]
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


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**

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

Value

A list with the following elements:

tauhat  A vector of the changepoints found.

Author

Dean Bodenham

References


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)
**detectCUSUMMean**

**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. Default 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 \( \mu \) and \( \sigma \) 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>The vector (stream) in which to detect change(s).</td>
</tr>
<tr>
<td>r</td>
<td>Control parameter for EWMA. Must be in range ([0, 1]). Default is ( r=0.25 ).</td>
</tr>
<tr>
<td>L</td>
<td>Control parameter for EWMA. Default is ( L=3.00 )</td>
</tr>
<tr>
<td>BL</td>
<td>The burn-in length. Default is ( BL=50 ).</td>
</tr>
<tr>
<td>multiple</td>
<td>Boolean to use to decide whether to detect multiple changes or only a single change. Default is ( \text{TRUE} ) (i.e. detect multiple changes).</td>
</tr>
</tbody>
</table>
**detectEWMAMean**

- **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 $\sqrt{\text{prechangeSigma}}$.

- **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 $\mu$ 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 $\sigma_{Z_j}$ is a scaled version of the in-control variance $\sigma$. 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

```r
# 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

```r
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 \( \text{true} \) (i.e. detect multiple changes).
- **single**: Boolean to use to decide whether to detect only a single change or multiple changes. Set to \( \text{!multiple} \), i.e. default is \( \text{false} \). If both \( \text{single} \) and \( \text{multiple} \) are set to \( \text{true} \), then only a single change will be detected; if both set to \( \text{false} \) then multiple changes will be detected (i.e. single dominates).
**detectFFMean**

**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**


**Examples**

```r
# 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 <- detectFFMean(x, alpha=0.01, lambda=0.95, BL=50, multiple=TRUE)

# now only a single (the first) changepoint
list_fff2 <- detectFFMean(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 <- detectFFMean(x, alpha=0.01, lambda=0.95, BL=50, single=TRUE,
                          prechangeMean=5, prechangeSigma=1)
```
ffstream  

**Description**
This function gives information on the package and how to see the vignette.

**Usage**

```r
ffstream()
```

ffstream_vignette  

**Description**
This function opens the ffstream vignette.

**Usage**

```r
ffstream_vignette()
```

Examples

```r
ffstream_vignette()
```

initAFFMean  

**Description**
This function makes it simple to initialise an AFF object.

**Usage**

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

**Arguments**

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

**Examples**

```r
library(Rcpp)
aff1 <- initAFFMean()  # initialises with `eta=0.01`
aff2 <- initAFFMean(0.1)  # initialises with `eta=0.1`
```
**initAFFMeanCD**  \hspace{1cm} *Initialisation of AFF change detector*

**Description**

This function makes it simple to initialise an FFF object.

**Usage**

\[
\text{initAFFMeanCD}(\alpha = 0.01, \eta = 0.01, BL = 50)
\]

**Arguments**

- **alpha**: The value of the significance level. Default value is 0.01, although it is recommended that the user set this parameter.
- **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 \(BL=50\). Must be at least greater than or equal to 2. No maximum. However, there is an exception: \(BL=0\) also works, but in this case the user needs to specify the \text{streamEstMean} and \text{streamEstSigma}; see

**Examples**

```
library(Rcpp)
affmeancd1 <- initAFFMeanCD()  # initialises with \(\alpha=0.01\),
                               # \(\eta=0.01\) and \(BL=50\)
affmeancd2 <- initAFFMeanCD(alpha=0.005, eta=0.1, BL=100)

affmeancd3 <- initAFFMeanCD(alpha=0.005, eta=0.1, BL=0)  # Example 3
affmeancd3$streamEstMean <- 0
affmeancd3$streamEstSigma <- 1
```

**initCUSUMMeanCD**  \hspace{1cm} *Initialisation of CUSUM*

**Description**

This function makes it simple to initialise a CUSUM object.

**Usage**

\[
\text{initCUSUMMeanCD}(k = 0.25, h = 8, BL = 50)
\]
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

```r
c1 <- initCUSUMMeanCD()  # initialises with $k=0.25$, $h=8.00$, $BL=50$
c2 <- initCUSUMMeanCD(k=0.5, h=4.00, BL=30)
```

Description

This function makes it simple to initialise an EWMA object.

Usage

```r
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 $BL=50$.

```r
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

Initialization of FFF

Description

This function makes it simple to initialise 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

Initialization of FFF mean change detector

Description

This function makes it simple to initialise an FFF object.

Usage

initFFFMeanCD(alpha = 0.01, lambda = 1, BL = 50)

Arguments

alpha The value of the significance level. Default value is 0.01, although it is recommended that the user set this parameter.
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() # initialises with alpha=0.01
fffcd2 <- initFFFMeanCD(0.05, 0.95) # initialises with alpha=0.05 and lambda=0.95 (and BL=50)
Index

computeAFFMean, 2, 3
computeFFFMean, 2, 3
demo_ffstream, 3
detectAFFMean, 5
detectCUSUMMean, 6
detectEWMAMean, 8
detectFFFMean, 10
ffstream, 12
ffstream_vignette, 12

initAFFMean, 12
initAFFMeanCD, 13
initCUSUMMeanCD, 13
initEWMAMeanCD, 14
initFFFMean, 15
initFFFMeanCD, 15