Package ‘CautiousLearning’

July 18, 2019

Type Package

Title Control Charts with Guaranteed In-Control Performance and Cautious Parameters Learning

Version 1.0.1

Date 2019-06-30

Author Giovanna Capizzi and Guido Masarotto

Maintainer Giovanna Capizzi <giovanna.capizzi@unipd.it>


License MIT + file LICENSE

Imports Rcpp (>= 1.0.0), spc

LinkingTo Rcpp, sitmo, BH

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-07-18 13:30:04 UTC

R topics documented:

CautiousLearning-package ............................................... 1
cautiousLearning ......................................................... 2
design ................................................................. 3
omp ................................................................. 5
simulation ............................................................ 6
CautiousLearning-package

Guaranteed In-Control Control Chart Performance with Cautious Parameter Learning

Description

Functions in this package allow to design, study and apply control charts based on the cautious parameter learning approach described in Capizzi and Masarotto (2019).

On system where the openMP standard is supported, these functions can take advantage of the computing power offered by a multicore workstation. See `omp` for the default setting.

Details

The package includes the following functions:

- Computation of the control limits via stochastic approximation: `x.cl`, `ewma.cl`, `cusum.cl`;
- Estimation errors and conditional run-length simulation: `ruv` and `rcrl`;
- Application to real data: `cautiousLearning`;
- Controlling the number of used openMP cores and the random number generator seeds: `hasOMP`, `setOMPTThreads` and `setSITMOSeeds`.

Author(s)

Giovanna Capizzi <giovanna.capizzi@unipd.it> and Guido Masarotto <guido.masarotto@unipd.it>

Maintainer: Giovanna Capizzi

References

Capizzi, G. and Masarotto, G. (2019) "Guaranteed In-Control Control Chart Performance with Cautious Parameter Learning", accepted for publication in Journal of Quality Technology, a copy of the paper can be obtained from the authors.

cautiousLearning

Applications of control charts based on the cautious learning approach

Description

This function applies and, optionally, plots a control chart based on the cautious learning approach described in Capizzi and Masarotto (2019).

Usage

```
cautiousLearning(chart, x, mu0, s0, plot = TRUE)
```
Arguments

chart list with the same elements as those returned by `x.cl, ewma.cl` and `cusum.cl`.
x numeric vector containing the Phase II data.
mu0, s0 estimates of the in-control mean and standard deviation obtained by the Phase I reference sample.
plot if TRUE the control statistics and the cautious control limits are plotted.

Value

The function returns (invisibly when `plot==TRUE`) a numeric matrix containing

- column 1 for X and EWMA, columns 1-2 for CUSUM control statistic[s]
- columns 2-4 for X and EWMA, columns 3-5 for CUSUM central line, lower and upper control limits
- columns 5-7 for X and EWMA, columns 6-8 for CUSUM “cautious” estimates of the mean, standard deviation and critical value, i.e., using the notation in Capizzi and Masarotto (2019), $\hat{\mu}_{i-d}$, $\hat{\sigma}_{i-d}$ and $L_{i-d}$.

Author(s)

Giovanna Capizzi and Guido Masarotto

References

Capizzi, G. and Masarotto, G. (2019) "Guaranteed In-Control Control Chart Performance with Cautious Parameter Learning", accepted for publication in *Journal of Quality Technology*, a copy of the paper can be obtained from the authors.

Examples

```r
## EWMA control chart (nominal ARL=500, initial estimates based on 100 in-control observations)
chart <- list(chart = "EWMA",
             lambda = 0.2,
             limit = c(Linf=3.187, Delta=0.427, A=1.5, B=50, m=100))
## Phase I estimates
set.seed(12345)
xic <- rnorm(100, 12, 3)
m0 <- mean(xic)
s0 <- sd(xic)
## Phase II observations (one sigma mean shift starting at i=501)
x <- c(rnorm(500, 12, 3), rnorm(50, 15, 3))
## Monitoring
y <- cautiousLearning(chart, x, m0, s0)
head(y)
tail(y)
```
Description

These functions compute the control limits of $X$ (x.cl), EWMA (ewma.cl) and CUSUM (cusum.cl) control charts based on the cautious learning approach. The stochastic approximation algorithm, described in the Appendix A of Capizzi and Masarotto (2019), is used.

When openMP is supported, computation can be distributed on multiple cores. See omp.

Usage

```
x.cl(m, arl0, alpha = 0.1, beta = 0.05, H = 200, A = 1.5, B = 50, Ninit = 1000, Nfinal = 30000)
ewma.cl(lambda, m, arl0, alpha = 0.1, beta = 0.05, H = 200, A = 1.5, B = 50, Ninit = 1000, Nfinal = 30000)
cusum.cl(k, m, arl0, alpha = 0.1, beta = 0.05, H = 200, A = 1.5, B = 50, Ninit = 1000, Nfinal = 30000)
```

Arguments

- **lambda**: EWMA smoothing constant.
- **k**: CUSUM reference value.
- **m**: number of in-control observations used to estimate the process mean and standard deviation at the beginning of the monitoring phase.
- **arl0, alpha, beta, H**: desired in-control average run-length and constants defining the empirical guaranteed in-control performance condition. See equations (2) and (6) in Capizzi and Masarotto (2019).
- **A, B**: constants controlling when the parameters estimate are updated. See equation (3) in Capizzi and Masarotto (2019). If $A=NA$ and $B=NA$, the no-learning control limits are computed.
- **Ninit, Nfinal**: number of iterations used in the stochastic approximation algorithm. See Capizzi and Masarotto (2019), Appendix A.

Value

A list with the following elements:

- **chart**: string describing the control chart ("X", "EWMA" or "CUSUM").
- **lambda**: EWMA smoothing constant (only when chart=="EWMA").
- **k**: CUSUM reference value (only when chart=="CUSUM").
limit numeric vector of length equal to five containing the constants defining the cautious learning control limits, i.e., $L_\infty$, $\Delta$, $A$, $B$, and $m$ (see equation (3) and (4) in Capizzi and Masarotto (2019)).

Author(s)

Giovanna Capizzi and Guido Masarotto

References

Capizzi, G. and Masarotto, G. (2019) "Guaranteed In-Control Control Chart Performance with Cautious Parameter Learning", accepted for publication in *Journal of Quality Technology*, a copy of the paper can be obtained from the authors.

Examples

```r
## Only for testing: the number of iterations is reduced
## to reduce the computing time
Ninit <- 50
Nfinal <- 100
H <- 50
x.cl(100, 500, Ninit=Ninit, Nfinal=Nfinal, H=H)
x.cl(100, 500, A=NA, B=NA, Ninit=Ninit, Nfinal=Nfinal, H=H)
ewma.cl(0.2, 100, 500, Ninit=Ninit, Nfinal=Nfinal, H=H)
cusum.cl(1, 100, 500, Ninit=Ninit, Nfinal=Nfinal, H=H)

## Using the default number of iterations
x.cl(100, 500)
x.cl(100, 500, A=NA, B=NA)
ewma.cl(0.2, 100, 500)
cusum.cl(1, 100, 500)
```

### Description

The functions can be used

- to check if the current system supports the openMP standard;
- to control the number of used cores;
- to set the seeds of the random number generators.

### Usage

```r
hasOMP ()
setOMPThreads(nthreads)
setSITMOSeeds(seed)
```
Arguments

- **nthreads**: number of OpenMP threads to be used.
- **seed**: number between 0 and 1 used to set the seeds of the random number generators in each threads.

Details

Each openMP thread (or the single thread used on systems where openMP is not supported) uses a separate *sitmo* random number generator. See *sitmo-package*.

Value

Function `hasOMP` returns TRUE/FALSE if the system supports/does not support openMP.

Functions `setOMPThreads` and `setSITMOSeeds` do not return any value.

Note

When the package is loaded, the following code is automatically executed

- if (hasOMP()) setOMPThreads(parallel::detectCores())
- setSITMOSeeds(runif(1))

Author(s)

Giovanna Capizzi and Guido Masarotto

---

**simulation**

*Estimation errors and conditional run-length simulation*

Description

Function `ruv` simulates the standardized estimation errors at the starting of the monitoring phase (see Section 2.3 of Capizzi and Masarotto (2019)).

Function `rcrl` simulates, under different in-control or out-control scenarios, the conditional run-length given the standardized estimation errors. When openMP is supported, computation can be distributed on multiple cores. See *omp*.

Usage

```r
ruv(n, m)
rcrl(n, chart, u, v, tau, delta, omega, maxrl = 1000000L)
```
Arguments

n
number of simulated values.

m
number of in-control observations available at the starting of the monitoring phase.

chart
list with the same elements as those returned by x.cl, ewma.cl and cusum.cl.

u, v
values of the estimation errors (scalars).

tau, delta, omega
when i<tau, observations are distributed as N(mu,sigma^2) random variables; when i>=tau, observations are distributed as N(mu+delta*sigma, (omega*sigma)^2) random variables.

maxrl
run-length are truncated at i=maxrl.

Value

ruv
numeric matrix of dimension nx2.

rcrl
integer vector of length n.

Author(s)

Giovanna Capizzi and Guido Masarotto

References

Capizzi, G. and Masarotto, G. (2019) "Guaranteed In-Control Control Chart Performance with Cautious Parameter Learning", accepted for publication in Journal of Quality Technology, a copy of the paper can be obtained from the authors.

Examples

ruv(5, 100)
## EWMA control chart (nominal ARL=500,
## initial estimates based on 100 in-control observations)
chart <- list(chart = "EWMA",
lambda = 0.2,
limit = c(Linf=3.187, Delta=0.427, A=1.5, B=50, m=100))
rcrl(10, chart, 2, 1, 50, 2, 1)