Package: mlAEWMA (via r-universe)

November 8, 2026

Type Package

Title Machine Learning Based Adaptive EWMA Control Charts for Process Dispersion

Version 0.1.0

Description Monitors the variance of a process with adaptive exponentially weighted moving average (AEWMA) control charts whose smoothing weight is predicted by a regression model fitted to the transformed subgroup variance. The subgroup variance is transformed as in Castagliola (2005) <doi:10.1002/qre.686>; the smoothing weight is predicted by one of nine pre-fitted models from four families (neural networks, support vector machines, regression trees and boosting, and a quantile regression with lagged inputs), following the parameter-free approach of Noor-ul-Amin et al. (2024) <doi:10.1038/s41598-024-82699-8>, related work on support vector regression based adaptive EWMA charts by Kazmi and Noor-ul-Amin (2024) <doi:10.1002/qre.3603>, and the adaptive EWMA idea of Capizzi and Masarotto (2003) <doi:10.1198/004017003000000023>. The comparison of the models is described in Atar, Purutçuoğlu and Bursalı (2026), accepted at the 3rd International Congress on Data Science and Information Technologies (INFTEC 2026). Control limits are calibrated by Monte Carlo simulation to a common in-control average run length of 370 for subgroups of size five. Functions are provided to run a chart on data, to compare all nine charts, and to estimate average run lengths by simulation.

License GPL (>= 2)

Encoding UTF-8

Depends R (>= 3.5.0)

Imports e1071, gbm, kernlab, nnet, parallel, qrnn, rpart, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Gizem Atar [aut, cre], Ahmet Bursalı [aut], Vilda Purutçuoğlu [aut]

Maintainer Gizem Atar <gizem.atar@ostimteknik.edu.tr>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-08 19:51:53 UTC

RemoteUrl https://github.com/cran/mlAEWMA

RemoteRef HEAD

RemoteSha d3f6dc9ccaa35276058f81b128fd09757db5281c

Contents

calculate_arl 2
mlaewma_compare 3
mlaewma_family 5
patient 9

Index **10**

`calculate_arl` *Average run length of an AEWMA chart by simulation*

Description

Estimates the average run length (ARL), its standard deviation (SDRL) and standard error for one of the nine charts, for subgroups drawn from a normal distribution with variance `shift` (in-control standard deviation 1, so `shift = 1` gives the in-control ARL0).

Usage

```
calculate_arl(
  method,
  h_value,
  shift = 1,
  subgroup_size = 5,
  nsim = 500,
  max_rl = 10000,
  seed = 42,
  n_cores = 1L,
  block = 500,
  n_streams = 3L
)
```

Arguments

method	One of "nn", "qrnn", "mcqrnn", "svr", "svqr", "rsvqr", "dt", "gb", "dqr".
h_value	Control limit.
shift	Variance multiplier of the simulated process (1 = in control).
subgroup_size	Subgroup size; only 5 is supported.
nsim	Number of simulated charts (at least 2).
max_rl	Maximum run length; longer runs are censored.
seed	Random seed.
n_cores	Number of worker processes (1 = no cluster). At most 2 are used while R CMD check runs.
block	Number of subgroups generated at a time (speed only; ignored for "dqr"). It does not change the results.
n_streams	Number of independent random number streams (see Details). 1 uses a single Mersenne-Twister stream.

Details

The result depends only on seed, n_streams and nsim. The simulation work is split into n_streams independent random number streams; with the default n_streams = 3 the numbers equal those used to calibrate the control limits. n_cores only sets how many worker processes run the streams and never changes the result. The random number generator state of the calling session is left unchanged.

Value

A named numeric vector with ARL, SDRL, ARL_SE and Censor_Rate (share of runs that reached max_rl).

Examples

```
calculate_arl("nn", h_value = 0.846, nsim = 30, seed = 1, n_streams = 1)
```

---

mlaewma_compare *Compare all nine AEWMA charts*

Description

Runs the neural network (nn, qrnn, mcqrnn), SVM (svr, svqr, rsvqr), tree (dt, gb) and DQR (dqr) charts on the same data and collects them in one table. With shifts, the out-of-control average run lengths are simulated and the models are ranked.

Usage

```
mlaewma_compare(
  data,
  sigma0 = NULL,
  subgroup_size = 5,
  h = NULL,
  find_h = FALSE,
  target_arl0 = 370,
  h_search_nsim = 3000,
  h_validation_nsim = 20000,
  arl_tolerance = 5,
  shifts = NULL,
  shift_nsim = 10000,
  shift_max_rl = 10000,
  nsim = 500,
  max_rl = 10000,
  seed = 42,
  n_cores = NULL,
  source_dir = ".",
  n_streams = 3L
)
```

Arguments

`data`	Numeric vector of observations (cut into subgroups of `subgroup_size`; an incomplete last subgroup is dropped) or a matrix or data frame with one subgroup per row.
`sigma0`	In-control standard deviation of the process. If `NULL`, the standard deviation of all observations is used.
`subgroup_size`	Subgroup size; only 5 is supported.
`h`	`NULL` for the calibrated defaults, otherwise a named vector or list that contains all nine model names (`nn`, `qrnn`, `mcqrnn`, `svr`, `svqr`, `rsvqr`, `dt`, `gb`, `dqr`; case-insensitive).
`find_h`	If `TRUE`, search `h` by bisection so that the in-control ARL equals `target_arl0` (slow).
`target_arl0`	Target in-control ARL for `find_h`.
`h_search_nsim`	Number of simulations per bisection step.
`h_validation_nsim`	Number of simulations for the final validation of the found `h`.
`arl_tolerance`	Bisection stops when the simulated ARL0 is within this distance of `target_arl0`.
`shifts`	Optional vector of variance multipliers for which the out-of-control ARL is simulated.
`shift_nsim`, `shift_max_rl`	Number of simulations and maximum run length used for `shifts`.
`nsim`, `max_rl`	Simulation settings used by `find_h` and `shifts` (`nsim` is not used for the application run).

seed	Random seed for simulations.
n_cores	Number of worker processes for simulation; NULL means 1.
source_dir	Not used any more; kept so that older calls still work.
n_streams	Number of independent random number streams used for simulation; the default 3 reproduces the calibration of the default control limits.

Value

A list with comparison (one row per model: Family, Method, h, First_Signal), models (the nine individual results), performance, rank_summary and family_results (the four family results).

Examples

```
data(patient)
mlaewma_compare(patient, sigma0 = 4.05)$comparison
```

---

mlaewma_family *AEWMA control charts by model family*

Description

Run the machine learning based adaptive EWMA chart for process dispersion on subgroup data. Each subgroup variance is transformed as in Castagliola (2005), a pre-fitted model predicts the smoothing weight theta (restricted to [0, 1]) and the chart statistic is updated as AEWMA_t = theta_t * D_t + (1 - theta_t) * AEWMA_(t-1). The first subgroup with |AEWMA_t| > h is the first signal.

Usage

```
mlaewma_neural(
  data,
  method = "qrnn",
  h = NULL,
  sigma0 = NULL,
  subgroup_size = 5,
  find_h = FALSE,
  target_arl0 = 370,
  h_search_nsim = 3000,
  h_validation_nsim = 20000,
  arl_tolerance = 5,
  shifts = NULL,
  shift_nsim = 10000,
  shift_max_rl = 10000,
  nsim = 500,
  max_rl = 10000,
  seed = 42,
  n_cores = 1L,
  n_streams = 3L
```

```
)

mlaewma_svm(
  data,
  method = "svqr",
  h = NULL,
  sigma0 = NULL,
  subgroup_size = 5,
  find_h = FALSE,
  target_arl0 = 370,
  h_search_nsim = 3000,
  h_validation_nsim = 20000,
  arl_tolerance = 5,
  shifts = NULL,
  shift_nsim = 10000,
  shift_max_rl = 10000,
  nsim = 500,
  max_rl = 10000,
  seed = 42,
  n_cores = 1L,
  n_streams = 3L
)

mlaewma_tree(
  data,
  method = "gb",
  h = NULL,
  sigma0 = NULL,
  subgroup_size = 5,
  find_h = FALSE,
  target_arl0 = 370,
  h_search_nsim = 3000,
  h_validation_nsim = 20000,
  arl_tolerance = 5,
  shifts = NULL,
  shift_nsim = 10000,
  shift_max_rl = 10000,
  nsim = 500,
  max_rl = 10000,
  seed = 42,
  n_cores = 1L,
  n_streams = 3L
)

mlaewma_dqr(
  data,
  method = "dqr",
  h = NULL,
```

```
  sigma0 = NULL,
  subgroup_size = 5,
  find_h = FALSE,
  target_arl0 = 370,
  h_search_nsim = 3000,
  h_validation_nsim = 20000,
  arl_tolerance = 5,
  shifts = NULL,
  shift_nsim = 10000,
  shift_max_rl = 10000,
  nsim = 500,
  max_rl = 10000,
  seed = 42,
  n_cores = 1L,
  n_streams = 3L
)
```

Arguments

`data`	Numeric vector of observations (cut into subgroups of `subgroup_size`; an incomplete last subgroup is dropped) or a matrix or data frame with one subgroup per row.
`method`	Model to use, or `"all"` for every model of the family.
`h`	Control limit. `NULL` uses the calibrated default. For `method = "all"` a vector with one value per model, named (case-insensitive) or in model order.
`sigma0`	In-control standard deviation of the process. If `NULL`, the standard deviation of all observations is used.
`subgroup_size`	Subgroup size; only 5 is supported.
`find_h`	If `TRUE`, search `h` by bisection so that the in-control ARL equals `target_arl0` (slow).
`target_arl0`	Target in-control ARL for `find_h`.
`h_search_nsim`	Number of simulations per bisection step.
`h_validation_nsim`	Number of simulations for the final validation of the found `h`.
`arl_tolerance`	Bisection stops when the simulated ARL0 is within this distance of `target_arl0`.
`shifts`	Optional vector of variance multipliers for which the out-of-control ARL is simulated.
`shift_nsim`, `shift_max_rl`	Number of simulations and maximum run length used for `shifts`.
`nsim`, `max_rl`	Simulation settings used by `find_h` and `shifts` (`nsim` is not used for the application run).
`seed`	Random seed for simulations.
`n_cores`	Number of worker processes used for simulation (default 1; at most 2 while `R CMD check` runs).
`n_streams`	Number of independent random number streams used for simulation; the default 3 reproduces the calibration of the default control limits.

Details

The four functions share all arguments. `mlaewma_neural()` offers `"nn"`, `"qrnn"` and `"mcqrnn"`; `mlaewma_svm()` offers `"svr"`, `"svqr"` and `"rsvqr"`; `mlaewma_tree()` offers `"dt"` and `"gb"`; `mlaewma_dqr()` has the single model `"dqr"`, whose smoothing weight also depends on the two previous values of `D`. With `method = "all"` every model of the family is run and compared (for the one-model DQR family `"all"` is the same as `"dqr"`). The `method` is not case sensitive. The models, the transformation constants and the control limits are calibrated for subgroups of size 5 only.

`first_signal` is the number of the first subgroup that signals, counted from the start of `data`; it is `NA` if the chart never signals.

Choosing `sigma0`. `sigma0` must be the standard deviation of the process while it is in control, estimated from a clean reference sample. If it is left `NULL`, the standard deviation of all supplied observations is used, which is inflated when `data` already contains an out-of-control stretch and can hide the shift.

Limitations of the models. The charts are calibrated for an in-control ARL of about 370 (within roughly 2 percent; the bisection tolerance is +/- 5). Their ability to detect changes differs a lot between models and is not guaranteed:

- `"qrnn"` and `"mcqrnn"` assign almost no weight to large positive `D`; their average run length for a moderate variance increase (variance multiplier 1.5) is longer than the in-control one. They are not suitable for monitoring variance increases.
- `"nn"`, `"dt"`, `"gb"` and `"dqr"` detect variance increases but practically not decreases. `"svr"`, `"svqr"` and `"rsvqr"` react to both directions.
- The models were trained on values of the transformed statistic between 0 and 1.9, but about half of the in-control values of `D` are negative, so the predicted weights are partly extrapolated.

See the vignette for simulated comparisons.

Simulation (needed when `find_h = TRUE` or `shifts` is given) is controlled by `n_streams` and `n_cores`; see `calculate_arl()`. The results depend on `seed`, `n_streams` and the number of simulations, not on `n_cores`.

Value

For a single model a list with `method`, `h`, `first_signal`, `results` (data frame with `Subgroup`, `Variance`, `D`, `Theta`, `AEWMA`, `h`, `Exceeds_Limit`), `calibration` (only with `find_h = TRUE`) and, if `shifts` is given, `performance`. For `method = "all"` a list with one such element per model plus `comparison`, `performance`, `best`, `rank_summary` and `calibration`.

References

Castagliola, P. (2005). A new S2-EWMA control chart for monitoring the process variance. *Quality and Reliability Engineering International*, 21(8), 781-794. doi:10.1002/qre.686

Noor-ul-Amin, M., Kazmi, M. W., Alkhalaf, S., Abdel-Khalek, S. and Nabi, M. (2024). Machine learning based parameter-free adaptive EWMA control chart to monitor process dispersion. *Scientific Reports*, 14, 31271. doi:10.1038/s41598024826998

Kazmi, M. W. and Noor-ul-Amin, M. (2024). Adaptive EWMA control chart by using support vector regression. *Quality and Reliability Engineering International*, 40(7), 3831-3843. doi:10.1002/qre.3603

Capizzi, G. and Masarotto, G. (2003). An adaptive exponentially weighted moving average control chart. *Technometrics*, 45(3), 199-207. doi:10.1198/004017003000000023

Atar, G., Purutçuoğlu, V. and Bursalı, A. (2026). Comparison of machine learning models in a parameter-free adaptive EWMA control chart for process dispersion monitoring. Accepted at the 3rd International Congress on Data Science and Information Technologies (INFTEC 2026).

Examples

```
data(patient)
res <- mlaewma_neural(patient, method = "nn", sigma0 = 4.05)
res$first_signal
head(res$results)
mlaewma_tree(patient, method = "all", sigma0 = 4.05)$comparison

# a variance increase after subgroup 20 (in-control sigma0 = 2)
set.seed(10)
x <- c(rnorm(100, sd = 2), rnorm(60, sd = 4))
mlaewma_svm(x, method = "svqr", sigma0 = 2)$first_signal
```

---

patient *Example subgroup data*

Description

Thirty subgroups of five measurements each (one subgroup per row) from a patient monitoring example. The in-control standard deviation used with this example is 4.05.

Usage

```
patient
```

Format

A numeric matrix with 30 rows and 5 columns.

References

Jones-Farmer, L. A. and Champ, C. W. (2010). A distribution-free phase I control chart for subgroup scale. *Journal of Quality Technology*, 42(4), 373-387. doi:10.1080/00224065.2010.11917834

Examples

```
data(patient)
dim(patient)
```

Index

∗ **datasets**
patient, 9

calculate_arl, 2
calculate_arl(), *8*

mlaewma_compare, 3
mlaewma_dqr *(*mlaewma_family*)*, 5
mlaewma_family, 5
mlaewma_neural *(*mlaewma_family*)*, 5
mlaewma_svm *(*mlaewma_family*)*, 5
mlaewma_tree *(*mlaewma_family*)*, 5

patient, 9