

# Package ‘samurais’

July 16, 2019

**Type** Package

**Title** Statistical Models for the Unsupervised Segmentation of Time Series

**Version** 0.1.0

**Description** samurais is a toolbox including many original and flexible user-friendly statistical latent variable models and efficient unsupervised algorithms to segment and represent time-series data (univariate or multivariate), and more generally, longitudinal data, which include regime changes. samurais is built upon the following packages, each of them is an autonomous time series segmentation approach: 'RHLP', 'HMMR', 'PWR', 'MRHLP', 'MHMMR'. For the advantages/differences of each of them, the user is referred to our mentioned paper references.

**URL** <https://github.com/fchamroukhi/SaMUraIS>

**License** GPL (>= 3)

**Depends** R (>= 2.10)

**Imports** methods,  
stats,  
MASS,  
Rcpp

**Suggests** knitr,  
rmarkdown

**LinkingTo** Rcpp,  
RcppArmadillo

**Collate** samurais-package.R  
RcppExports.R  
utils.R  
dynamicProg.R  
fitPWRFisher.R  
mkStochastic.R  
hmmProcess.R  
MData.R  
ParamHMMR.R  
ParamMHMMR.R  
ParamMRHLP.R  
ParamPWR.R  
ParamRHLP.R  
StatHMMR.R  
StatMHMMR.R

StatMRHLP.R  
 StatPWR.R  
 StatRHLP.R  
 ModelHMMR.R  
 ModelMHMMR.R  
 ModelMRHLP.R  
 ModelPWR.R  
 ModelRHLP.R  
 emHMMR.R  
 emMHMMR.R  
 emMRHLP.R  
 emRHLP.R  
 selectHMMR.R  
 selectMHMMR.R  
 selectMRHLP.R  
 selectRHLP.R  
 data-multivrealdataset.R  
 data-multivtoydataset.R  
 data-univrealdataset.R  
 data-univtoydataset.R

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 6.1.1

## R topics documented:

samurais-package . . . . .	3
emHMMR . . . . .	4
emMHMMR . . . . .	5
emMRHLP . . . . .	6
emRHLP . . . . .	7
fitPWRFisher . . . . .	8
hmmProcess . . . . .	9
MData-class . . . . .	10
mkStochastic . . . . .	10
ModelHMMR-class . . . . .	11
ModelMHMMR-class . . . . .	11
ModelMRHLP-class . . . . .	12
ModelPWR-class . . . . .	13
ModelRHLP-class . . . . .	13
multivrealdataset . . . . .	14
multivtoydataset . . . . .	15
ParamHMMR-class . . . . .	16
ParamMHMMR-class . . . . .	17
ParamMRHLP-class . . . . .	18
ParamPWR-class . . . . .	19
ParamRHLP-class . . . . .	19
selectHMMR . . . . .	20

selectMHMMR . . . . .	21
selectMRHLP . . . . .	22
selectRHLP . . . . .	23
StatHMMR-class . . . . .	24
StatMHMMR-class . . . . .	25
StatMRHLP-class . . . . .	26
StatPWR-class . . . . .	27
StatRHLP-class . . . . .	28
univrealdataset . . . . .	29
univtoydataset . . . . .	30

<b>Index</b>	<b>31</b>
--------------	-----------

---

samurais-package	<i>SaMURaiS: Statistical Models for the Unsupervised segmentation of time Series</i>
------------------	--------------------------------------------------------------------------------------

---

## Description

samurais is a toolbox including many original and flexible user-friendly statistical latent variable models and efficient unsupervised algorithms to segment and represent time-series data (univariate or multivariate), and more generally, longitudinal data, which include regime changes.

samurais contains the following time series segmentation models:

- RHLP;
- HMM/HMMR;
- PWR;
- MRHLP;
- MHMMR;

For the advantages/differences of each of them, the user is referred to our mentioned paper references.

To learn more about samurais, start with the vignettes: `browseVignettes(package = "samurais")`

## Author(s)

**Maintainer:** Florian Lecocq <florian.lecocq@outlook.com>

Authors:

- Faicel Chamroukhi <faicel.chamroukhi@unicaen.fr> (0000-0002-5894-3103)
- Marius Bartcus <marius.bartcus@gmail.com>

## References

Chamroukhi, F., and Hien D. Nguyen. 2019. *Model-Based Clustering and Classification of Functional Data*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. <https://doi.org/10.1002/widm.1298>.

Chamroukhi, F. 2015. *Statistical Learning of Latent Data Models for Complex Data Analysis*. Habilitation Thesis (HDR), Universite de Toulon. <https://chamroukhi.com/Dossier/FChamroukhi-Habilitation.pdf>.

Trabelsi, D., S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. 2013. *An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression*. IEEE Transactions on Automation Science and Engineering 3 (10): 829–335. <https://chamroukhi.com/papers/Chamroukhi-MHMMR-IeeeTase.pdf>.

Chamroukhi, F., D. Trabelsi, S. Mohammed, L. Oukhellou, and Y. Amirat. 2013. *Joint Segmentation of Multivariate Time Series with Hidden Process Regression for Human Activity Recognition*. Neurocomputing 120: 633–44. [https://chamroukhi.com/papers/chamroukhi\\_et\\_al\\_neucomp2013b.pdf](https://chamroukhi.com/papers/chamroukhi_et_al_neucomp2013b.pdf).

Chamroukhi, F., A. Same, G. Govaert, and P. Aknin. 2010. *A Hidden Process Regression Model for Functional Data Description. Application to Curve Discrimination*. Neurocomputing 73 (7-9): 1210–21. [https://chamroukhi.com/papers/chamroukhi\\_neucomp\\_2010.pdf](https://chamroukhi.com/papers/chamroukhi_neucomp_2010.pdf).

Chamroukhi, F. 2010. *Hidden Process Regression for Curve Modeling, Classification and Tracking*. Ph.D. Thesis, Universite de Technologie de Compiegne. <https://chamroukhi.com/papers/FChamroukhi-Thesis.pdf>.

Chamroukhi, F., A. Same, G. Govaert, and P. Aknin. 2009. *Time Series Modeling by a Regression Approach Based on a Latent Process*. Neural Networks 22 (5-6): 593–602. [https://chamroukhi.com/papers/Chamroukhi\\_Neural\\_Networks\\_2009.pdf](https://chamroukhi.com/papers/Chamroukhi_Neural_Networks_2009.pdf).

## See Also

Useful links:

- <https://github.com/fchamroukhi/SaMUraIS>

---

emHMMR	<i>emHMMR implements the EM (Baum-Welch) algorithm to fit a HMMR model.</i>
--------	-----------------------------------------------------------------------------

---

## Description

emHMMR implements the maximum-likelihood parameter estimation of the HMMR model by the Expectation-Maximization (EM) algorithm, known as Baum-Welch algorithm in the context of HMMs.

## Usage

```
emHMMR(X, Y, K, p = 3, variance_type = c("heteroskedastic",
    "homoskedastic"), n_tries = 1, max_iter = 1500, threshold = 1e-06,
    verbose = FALSE)
```

## Arguments

X	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
Y	Numeric vector of length $m$ representing the observed response/output $y_1, \dots, y_m$ .
K	The number of regimes/segments (HMMR components).
p	Optional. The order of the polynomial regression. By default, p is set at 3.
variance_type	Optional character indicating if the model is "homoskedastic" or "heteroskedastic" (i.e same variance or different variances for each of the K regmies). By default the model is "heteroskedastic".

n_tries	Optional. Number of runs of the EM algorithm. The solution providing the highest log-likelihood will be returned. If <code>n_tries &gt; 1</code> , then for the first run, parameters are initialized by uniformly segmenting the data into $K$ segments, and for the next runs, parameters are initialized by randomly segmenting the data into $K$ contiguous segments.
max_iter	Optional. The maximum number of iterations for the EM algorithm.
threshold	Optional. A numeric value specifying the threshold for the relative difference of log-likelihood between two steps of the EM as stopping criteria.
verbose	Optional. A logical value indicating whether or not values of the log-likelihood should be printed during EM iterations.

### Details

emMHMMR function implements the EM algorithm for the HMMR model. This function starts with an initialization of the parameters done by the method `initParam` of the class [ParamHMMR](#), then it alternates between the E-Step (method of the class [StatHMMR](#)) and the M-Step (method of the class [ParamHMMR](#)) until convergence (until the relative variation of log-likelihood between two steps of the EM algorithm is less than the threshold parameter).

### Value

EM returns an object of class [ModelHMMR](#).

### See Also

[ModelHMMR](#), [ParamHMMR](#), [StatHMMR](#)

---

emMHMMR	<i>emMHMMR implements the EM (Baum-Welch) algorithm to fit a MH-MMR model.</i>
---------	--------------------------------------------------------------------------------

---

### Description

emMHMMR implements the maximum-likelihood parameter estimation of the MHMMR model by the Expectation-Maximization (EM) algorithm, known as Baum-Welch algorithm in the context of HMMs.

### Usage

```
emMHMMR(X, Y, K, p = 3, variance_type = c("heteroskedastic",
    "homoskedastic"), n_tries = 1, max_iter = 1500, threshold = 1e-06,
    verbose = FALSE)
```

### Arguments

X	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
Y	Matrix of size $(m, d)$ representing a $d$ dimension time series observed at points $1, \dots, m$ .
K	The number of regimes (MHMMR components).
p	Optional. The order of the polynomial regression. By default, $p$ is set at 3.

variance_type	Optional character indicating if the model is "homoskedastic" or "heteroskedastic". By default the model is "heteroskedastic".
n_tries	Optional. Number of runs of the EM algorithm. The solution providing the highest log-likelihood will be returned. If $n\_tries > 1$ , then for the first run, parameters are initialized by uniformly segmenting the data into $K$ segments, and for the next runs, parameters are initialized by randomly segmenting the data into $K$ contiguous segments.
max_iter	Optional. The maximum number of iterations for the EM algorithm.
threshold	Optional. A numeric value specifying the threshold for the relative difference of log-likelihood between two steps of the EM as stopping criteria.
verbose	Optional. A logical value indicating whether or not values of the log-likelihood should be printed during EM iterations.

### Details

emMHMMR function implements the EM algorithm. This function starts with an initialization of the parameters done by the method `initParam` of the class [ParamMHMMR](#), then it alternates between the E-Step (method of the class [StatMHMMR](#)) and the M-Step (method of the class [ParamMHMMR](#)) until convergence (until the relative variation of log-likelihood between two steps of the EM algorithm is less than the threshold parameter).

### Value

EM returns an object of class [ModelMHMMR](#).

### See Also

[ModelMHMMR](#), [ParamMHMMR](#), [StatMHMMR](#)

---

emMRHLP

---

*emMRHLP implements the EM algorithm to fit a MRHLP model.*


---

### Description

emMRHLP implements the maximum-likelihood parameter estimation of the MRHLP model by the Expectation-Maximization (EM) algorithm.

### Usage

```
emMRHLP(X, Y, K, p = 3, q = 1, variance_type = c("heteroskedastic",
"homoskedastic"), n_tries = 1, max_iter = 1500, threshold = 1e-06,
verbose = FALSE, verbose_IRLS = FALSE)
```

### Arguments

X	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
Y	Matrix of size $(m, d)$ representing a $d$ dimension function of $X$ observed at points $1, \dots, m$ . $Y$ is the observed response/output.
K	The number of regimes (MRHLP components).

p	Optional. The order of the polynomial regression. By default, p is set at 3.
q	Optional. The dimension of the logistic regression. For the purpose of segmentation, it must be set to 1 (which is the default value).
variance_type	Optional character indicating if the model is "homoskedastic" or "heteroskedastic". By default the model is "heteroskedastic".
n_tries	Optional. Number of runs of the EM algorithm. The solution providing the highest log-likelihood will be returned. If n_tries > 1, then for the first run, parameters are initialized by uniformly segmenting the data into K segments, and for the next runs, parameters are initialized by randomly segmenting the data into K contiguous segments.
max_iter	Optional. The maximum number of iterations for the EM algorithm.
threshold	Optional. A numeric value specifying the threshold for the relative difference of log-likelihood between two steps of the EM as stopping criteria.
verbose	Optional. A logical value indicating whether or not values of the log-likelihood should be printed during EM iterations.
verbose_IRLS	Optional. A logical value indicating whether or not values of the criterion optimized by IRLS should be printed at each step of the EM algorithm.

### Details

emMRHLP function implements the EM algorithm of the MRHLP model. This function starts with an initialization of the parameters done by the method `initParam` of the class [ParamMRHLP](#), then it alternates between the E-Step (method of the class [StatMRHLP](#)) and the M-Step (method of the class [ParamMRHLP](#)) until convergence (until the relative variation of log-likelihood between two steps of the EM algorithm is less than the threshold parameter).

### Value

EM returns an object of class [ModelMRHLP](#).

### See Also

[ModelMRHLP](#), [ParamMRHLP](#), [StatMRHLP](#)

---

emRHLP

---

*emRHLP implements the EM algorithm to fit a RHLP model.*


---

### Description

emRHLP implements the maximum-likelihood parameter estimation of the RHLP model by the Expectation-Maximization (EM) algorithm.

### Usage

```
emRHLP(X, Y, K, p = 3, q = 1, variance_type = c("heteroskedastic",
  "homoskedastic"), n_tries = 1, max_iter = 1500, threshold = 1e-06,
  verbose = FALSE, verbose_IRLS = FALSE)
```

**Arguments**

X	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
Y	Numeric vector of length $m$ representing the observed response/output $y_1, \dots, y_m$ .
K	The number of regimes (RHLP components).
p	Optional. The order of the polynomial regression. By default, p is set at 3.
q	Optional. The dimension of the logistic regression. For the purpose of segmentation, it must be set to 1 (which is the default value).
variance_type	Optional character indicating if the model is "homoskedastic" or "heteroskedastic". By default the model is "heteroskedastic".
n_tries	Optional. Number of runs of the EM algorithm. The solution providing the highest log-likelihood will be returned. If $n\_tries > 1$ , then for the first run, parameters are initialized by uniformly segmenting the data into K segments, and for the next runs, parameters are initialized by randomly segmenting the data into K contiguous segments.
max_iter	Optional. The maximum number of iterations for the EM algorithm.
threshold	Optional. A numeric value specifying the threshold for the relative difference of log-likelihood between two steps of the EM as stopping criteria.
verbose	Optional. A logical value indicating whether or not values of the log-likelihood should be printed during EM iterations.
verbose_IRLS	Optional. A logical value indicating whether or not values of the criterion optimized by IRLS should be printed at each step of the EM algorithm.

**Details**

emRHLP function implements the EM algorithm for the RHLP model. This function starts with an initialization of the parameters done by the method `initParam` of the class [ParamRHLP](#), then it alternates between the E-Step (method of the class [StatRHLP](#)) and the M-Step (method of the class [ParamRHLP](#)) until convergence (until the relative variation of log-likelihood between two steps of the EM algorithm is less than the threshold parameter).

**Value**

EM returns an object of class [ModelRHLP](#).

**See Also**

[ModelRHLP](#), [ParamRHLP](#), [StatRHLP](#)

---

fitPWRFisher

---

*fitPWRFisher implements an optimized dynamic programming algorithm to fit a PWR model.*


---

**Description**

fitPWRFisher is used to fit a Piecewise Regression (PWR) model by maximum-likelihood via an optimized dynamic programming algorithm. The estimation performed by the dynamic programming algorithm provides an optimal segmentation of the time series.



**Usage**

```
fitPWRFisher(X, Y, K, p = 3)
```

**Arguments**

X                      Numeric vector of length  $m$  representing the covariates/inputs  $x_1, \dots, x_m$ .  
Y                      Numeric vector of length  $m$  representing the observed response/output  $y_1, \dots, y_m$ .  
K                      The number of regimes/segments (PWR components).  
p                      Optional. The order of the polynomial regression. By default, p is set at 3.

**Details**

fitPWRFisher function implements an optimized dynamic programming algorithm of the PWR model. This function starts with the calculation of the "cost matrix" then it estimates the transition points given K the number of regimes thanks to the method computeDynamicProgram (method of the class [ParamPWR](#)).

**Value**

fitPWRFisher returns an object of class [ModelPWR](#).

**See Also**

[ModelPWR](#), [ParamPWR](#), [StatPWR](#)

---

hmmProcess	<i>hmmProcess calculates the probability distribution of a random process following a Markov chain</i>
------------	--------------------------------------------------------------------------------------------------------

---

**Description**

hmmProcess calculates the probability distribution of a random process following a Markov chain

**Usage**

```
hmmProcess(prior, trans_mat, n)
```

**Arguments**

prior                      Numeric vector or a one row matrix of length K representing the prior probabilities of the Markov chain.  
trans\_mat                  Matrix of size  $(K, K)$  representing the transition matrix of the Markov chain.  
n                          Numeric. Number of variables of the Markov chain.

**Details**

hmmProcess calculates the distribution  $P(Z_1, \dots, Z_n; \pi, A)$  of a Markov chain  $(Z_1, \dots, Z_n)$  with prior probability  $\pi$  and transition matrix  $A$ .

The calculation is based on the following formula:

$$P(Z_i = k) = \sum_l P(Z_i = k, Z_{i-1} = l) = \sum_l P(Z_i = k | Z_{i-1} = l) \times P(Z_{i-1} = l) = \sum_l A_{lk} \times P(Z_{i-1})$$

**Value**

Matrix of size  $(n, K)$  giving the distribution of process given the  $K$ -state Markov chain parameters.

---

MData-class	<i>A Reference Class which represents multivariate data.</i>
-------------	--------------------------------------------------------------

---

**Description**

MData is a reference class which represents multivariate objects. The data can be ordered by time (multivariate time series). In the last case, the field  $X$  represents the time.

**Fields**

$X$  Numeric vector of length  $m$ .

$Y$  Matrix of size  $(m, d)$  representing a  $d$  dimension function of  $X$  observed at points  $1, \dots, m$ .

---

mkStochastic	<i>mkStochastic ensures that it is a stochastic vector, matrix or array.</i>
--------------	------------------------------------------------------------------------------

---

**Description**

mkStochastic ensures that it is a stochastic vector, matrix or array.

**Usage**

```
mkStochastic(M)
```

**Arguments**

$M$  A vector, matrix or array to transform.

**Details**

mkStochastic ensures that the giving argument is a stochastic vector, matrix or array, i.e., that the sum over the last dimension is 1.

**Value**

A vector, matrix or array for which the sum over the last dimension is 1.

---

ModelHMMR-class	<i>A Reference Class which represents a fitted HMMR model.</i>
-----------------	----------------------------------------------------------------

---

**Description**

ModelHMMR represents an estimated HMMR model.

**Fields**

param An object of class [ParamHMMR](#). It contains the estimated values of the parameters.

stat An object of class [StatHMMR](#). It contains all the statistics associated to the HMMR model.

**Methods**

plot(what = c("predicted", "filtered", "smoothed", "regressors", "loglikelihood"), ...)  
Plot method.

what The type of graph requested:

- "predicted" = Predicted time series and predicted regime probabilities (fields predicted and predict\_prob of class [StatHMMR](#)).
- "filtered" = Filtered time series and filtering regime probabilities (fields filtered and filter\_prob of class [StatHMMR](#)).
- "smoothed" = Smoothed time series, and segmentation (fields smoothed and klas of the class [StatHMMR](#)).
- "regressors" = Polynomial regression components (fields regressors and tau\_tk of class [StatHMMR](#)).
- "loglikelihood" = Value of the log-likelihood for each iteration (field stored\_loglik of class [StatHMMR](#)).

... Other graphics parameters.

By default, all the graphs mentioned above are produced.

summary(digits = getOption("digits")) Summary method.

digits The number of significant digits to use when printing.

**See Also**

[ParamHMMR](#), [StatHMMR](#)

---

ModelMHMMR-class	<i>A Reference Class which represents a fitted MHMMR model.</i>
------------------	-----------------------------------------------------------------

---

**Description**

ModelMHMMR represents an estimated MHMMR model.

**Fields**

param A [ParamMHMMR](#) object. It contains the estimated values of the parameters.

stat A [StatMHMMR](#) object. It contains all the statistics associated to the MHMMR model.

## Methods

`plot(what = c("predicted", "filtered", "smoothed", "regressors", "loglikelihood"), ...)` Plot method.

`what` The type of graph requested:

- "predicted" = Predicted time series and predicted regime probabilities (fields `predicted` and `predict_prob` of class [StatMHMMR](#)).
- "filtered" = Filtered time series and filtering regime probabilities (fields `filtered` and `filter_prob` of class [StatMHMMR](#)).
- "smoothed" = Smoothed time series, and segmentation (fields `smoothed` and `klas` of class [StatMHMMR](#)).
- "regressors" = Polynomial regression components (fields `regressors` and `tau_tk` of class [StatMHMMR](#)).
- "loglikelihood" = Value of the log-likelihood for each iteration (field `stored_loglik` of class [StatMHMMR](#)).

`...` Other graphics parameters.

By default, all the above graphs are produced.

`summary(digits = getOption("digits"))` Summary method.

`digits` The number of significant digits to use when printing.

## See Also

[ParamMHMMR](#), [StatMHMMR](#)

---

ModelMRHLP-class	<i>A Reference Class which represents a fitted MRHLP model.</i>
------------------	-----------------------------------------------------------------

---

## Description

ModelMRHLP represents an estimated MRHLP model.

## Fields

`param` A [ParamMRHLP](#) object. It contains the estimated values of the parameters.

`stat` A [StatMRHLP](#) object. It contains all the statistics associated to the MRHLP model.

## Methods

`plot(what = c("regressors", "estimatedsignal", "loglikelihood"), ...)` Plot method.

`what` The type of graph requested:

- "regressors" = Polynomial regression components (fields `polynomials` and `pi_ik` of class [StatMRHLP](#)).
- "estimatedsignal" = Estimated signal (fields `Ex` and `klas` of class [StatMRHLP](#)).
- "loglikelihood" = Value of the log-likelihood for each iteration (field `stored_loglik` of class [StatMRHLP](#)).

`...` Other graphics parameters.

By default, all the graphs mentioned above are produced.

`summary(digits = getOption("digits"))` Summary method.

`digits` The number of significant digits to use when printing.

**See Also**

[ParamMRHLP](#), [StatMRHLP](#)

---

ModelPWR-class

*A Reference Class which represents a fitted PWR model.*


---

**Description**

ModelPWR represents an estimated PWR model.

**Fields**

param A [ParamPWR](#) object. It contains the estimated values of the parameters.

stat A [StatPWR](#) object. It contains all the statistics associated to the PWR model.

**Methods**

plot(what = c("regressors", "segmentation"), ...) Plot method.

what The type of graph requested:

- "regressors" = Polynomial regression components (field regressors of class [StatPWR](#)).
- "segmentation" = Estimated signal (field mean\_function of class [StatPWR](#)).

... Other graphics parameters.

By default, all the graphs mentioned above are produced.

summary(digits = getOption("digits")) Summary method.

digits The number of significant digits to use when printing.

**See Also**

[ParamPWR](#), [StatPWR](#)

---

ModelRHLP-class

*A Reference Class which represents a fitted RHLP model.*


---

**Description**

ModelRHLP represents an estimated RHLP model.

**Fields**

param A [ParamRHLP](#) object. It contains the estimated values of the parameters.

stat A [StatRHLP](#) object. It contains all the statistics associated to the RHLP model.

## Methods

`plot(what = c("regressors", "estimatedsignal", "loglikelihood"), ...)` Plot method.

`what` The type of graph requested:

- "regressors" = Polynomial regression components (fields `polynomials` and `pi_ik` of class [StatRHLP](#)).
- "estimatedsignal" = Estimated signal (fields `Ex` and `klas` of class [StatRHLP](#)).
- "loglikelihood" = Value of the log-likelihood for each iteration (field `stored_loglik` of class [StatRHLP](#)).

`...` Other graphics parameters.

By default, all the graphs mentioned above are produced.

`summary(digits = getOption("digits"))` Summary method.

`digits` The number of significant digits to use when printing.

## See Also

[ParamRHLP](#), [StatRHLP](#)

---

multivrealdataset	<i>Time series representing the three acceleration components recorded over time with body mounted accelerometers during the activity of a given person.</i>
-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

---

## Description

This dataset is provided for illustration only and represents the three acceleration components recorded over time with body mounted accelerometers during the activity of a given person. These data consist therefore of multidimensional time series with several regime changes over time, each regime is associated with an activity.

## Usage

```
multivrealdataset
```

## Format

A data frame with 2253 rows and 4 columns:

**x** The covariate variable (the sampling time).

**y1** X axis component of the acceleration.

**y2** Y axis component of the acceleration.

**y3** Z axis component of the acceleration.

---

multivtoydataset	<i>A simulated non-stationary multidimensional time series with regime changes.</i>
------------------	-------------------------------------------------------------------------------------

---

### Description

A simulated non-stationary multidimensional time series with five regimes (segments). This time series is used for illustration.

### Usage

```
multivtoydataset
```

### Format

A data frame with 670 rows and 4 columns:

**x** The covariate variable (the sampling time for time series).

**y1** The first dimension of the time series. The latter has been generated as follows:

- First regime: 100 values of standard Normally distributed random numbers.
- Second regime: 120 values of Normally distributed random numbers with mean 7 and unit variance.
- Third regime: 200 values of Normally distributed random numbers with mean 4 and unit variance.
- Fourth regime: 100 values of Normally distributed random numbers with mean -1 and unit variance.
- Fifth regime: 150 values of Normally distributed random numbers with mean 3.5 and unit variance.

**y2** The second dimension of the time series. The latter has been generated as follows:

- First regime: 100 values of Normally distributed random numbers with mean 1 and unit variance.
- Second regime: 120 values of Normally distributed random numbers with mean 5 and unit variance.
- Third regime: 200 values of Normally distributed random numbers with mean 6 and unit variance.
- Fourth regime: 100 values of Normally distributed random numbers with mean -2 and unit variance.
- Fifth regime: 150 values of Normally distributed random numbers with mean 2 and unit variance.

**y3** The third dimension of the time series. The latter has been generated as follows:

- First regime: 100 values of Normally distributed random numbers with mean -2 and unit variance.
- Second regime: 120 values of Normally distributed random numbers with mean 10 and unit variance.
- Third regime: 200 values of Normally distributed random numbers with mean 8 and unit variance.
- Fourth regime: 100 values of Normally distributed random numbers and unit variance.
- Fifth regime: 150 values of Normally distributed random numbers with mean 5 and unit variance.

---

ParamHMMR-class	<i>A Reference Class which contains parameters of a HMMR model.</i>
-----------------	---------------------------------------------------------------------

---

## Description

ParamHMMR contains all the parameters of a HMMR model. The parameters are calculated by the initialization Method and then updated by the Method implementing the M-Step of the EM algorithm.

## Fields

- X Numeric vector of length  $m$  representing the covariates/inputs  $x_1, \dots, x_m$ .
- Y Numeric vector of length  $m$  representing the observed response/output  $y_1, \dots, y_m$ .
- m Numeric. Length of the response/output vector Y.
- K The number of regimes (HMMR components).
- p The order of the polynomial regression.
- variance\_type Character indicating if the model is homoskedastic (variance\_type = "homoskedastic") or heteroskedastic (variance\_type = "heteroskedastic"). By default the model is heteroskedastic.
- prior The prior probabilities of the Markov chain. prior is a row matrix of dimension  $(1, K)$ .
- trans\_mat The transition matrix of the Markov chain. trans\_mat is a matrix of dimension  $(K, K)$ .
- mask Mask applied to the transition matrices trans\_mat. By default, a mask of order one is applied.
- beta Parameters of the polynomial regressions.  $\beta = (\beta_1, \dots, \beta_K)$  is a matrix of dimension  $(p + 1, K)$ , with p the order of the polynomial regression. p is fixed to 3 by default.
- sigma2 The variances for the K regimes. If HMMR model is heteroskedastic (variance\_type = "heteroskedastic") then sigma2 is a matrix of size  $(K, 1)$  (otherwise HMMR model is homoskedastic (variance\_type = "homoskedastic") and sigma2 is a matrix of size  $(1, 1)$ ).
- nu The degree of freedom of the HMMR model representing the complexity of the model.
- phi A list giving the regression design matrices for the polynomial and the logistic regressions.

## Methods

- initParam(try\_algo = 1) Method to initialize parameters mask, prior, trans\_mat, beta and sigma2.  
If try\_algo = 1 then beta and sigma2 are initialized by segmenting the time series Y uniformly into K contiguous segments. Otherwise, beta and sigma2 are initialized by segmenting randomly the time series Y into K segments.
- MStep(statHMMR) Method which implements the M-step of the EM algorithm to learn the parameters of the HMMR model based on statistics provided by the object statHMMR of class [StatHMMR](#) (which contains the E-step).



---

ParamMHMMR-class	<i>A Reference Class which contains parameters of a MHMMR model.</i>
------------------	----------------------------------------------------------------------

---

## Description

ParamMHMMR contains all the parameters of a MHMMR model. The parameters are calculated by the initialization Method and then updated by the Method implementing the M-Step of the EM algorithm.

## Fields

- mData [MData](#) object representing the sample (covariates/inputs X and observed multivariate responses/outputs Y).
- K The number of regimes (MHMMR components).
- p The order of the polynomial regression.
- variance\_type Character indicating if the model is homoskedastic (variance\_type = "homoskedastic") or heteroskedastic (variance\_type = "heteroskedastic"). By default the model is heteroskedastic.
- prior The prior probabilities of the Markov chain. prior is a row matrix of dimension  $(1, K)$ .
- trans\_mat The transition matrix of the Markov chain. trans\_mat is a matrix of dimension  $(K, K)$ .
- mask Mask applied to the transition matrices trans\_mat. By default, a mask of order one is applied.
- beta Parameters of the polynomial regressions.  $\beta = (\beta_1, \dots, \beta_K)$  is an array of dimension  $(p + 1, d, K)$ , with p the order of the polynomial regression. p is fixed to 3 by default.
- sigma2 The variances for the K regimes. If MRHLP model is heteroskedastic (variance\_type = "heteroskedastic") then sigma2 is an array of size  $(d, d, K)$  (otherwise MRHLP model is homoskedastic (variance\_type = "homoskedastic") and sigma2 is a matrix of size  $(d, d)$ ).
- nu The degree of freedom of the MHMMR model representing the complexity of the model.
- phi A list giving the regression design matrices for the polynomial and the logistic regressions.

## Methods

- initParam(try\_algo = 1) Method to initialize parameters prior, trans\_mat, beta and sigma2.  
If try\_algo = 1 then beta and sigma2 are initialized by segmenting the time series Y uniformly into K contiguous segments. Otherwise, beta and sigma2 are initialized by segmenting randomly the time series Y into K segments.
- MStep(statMHMMR) Method which implements the M-step of the EM algorithm to learn the parameters of the MHMMR model based on statistics provided by the object statMHMMR of class [StatMHMMR](#) (which contains the E-step).

---

ParamMRHLP-class	<i>A Reference Class which contains the parameters of a MRHLP model.</i>
------------------	--------------------------------------------------------------------------

---

## Description

ParamMRHLP contains all the parameters of a MRHLP model. The paramerts are calculated by the initialization Method and then updated by the Method implementing the M-Step of the EM algorithm.

## Fields

- mData [MData](#) object representing the sample (covariates/inputs X and observed responses/outputs Y).
- K The number of regimes (MRHLP components).
- p The order of the polynomial regression.
- q The dimension of the logistic regression. For the purpose of segmentation, it must be set to 1.
- variance\_type Character indicating if the model is homoskedastic (variance\_type = "homoskedastic") or heteroskedastic (variance\_type = "heteroskedastic"). By default the model is heteroskedastic.
- W Parameters of the logistic process.  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_{K-1})$  is a matrix of dimension  $(q+1, K-1)$ , with q the order of the logistic regression. q is fixed to 1 by default.
- beta Parameters of the polynomial regressions.  $\beta = (\beta_1, \dots, \beta_K)$  is an array of dimension  $(p+1, d, K)$ , with p the order of the polynomial regression. p is fixed to 3 by default.
- sigma2 The variances for the K regimes. If MRHLP model is heteroskedastic (variance\_type = "heteroskedastic") then sigma2 is an array of size  $(d, d, K)$  (otherwise MRHLP model is homoskedastic (variance\_type = "homoskedastic") and sigma2 is a matrix of size  $(d, d)$ ).
- nu The degree of freedom of the MRHLP model representing the complexity of the model.
- phi A list giving the regression design matrices for the polynomial and the logistic regressions.

## Methods

- initParam(try\_algo = 1) Method to initialize parameters W, beta and sigma2.  
If try\_algo = 1 then beta and sigma2 are initialized by segmenting the time series Y uniformly into K contiguous segments. Otherwise, W, beta and sigma2 are initialized by segmenting randomly the time series Y into K segments.
- MStep(statMRHLP, verbose\_IRLS) Method which implements the M-step of the EM algorithm to learn the parameters of the MRHLP model based on statistics provided by the object statMRHLP of class [StatMRHLP](#) (which contains the E-step).

---

ParamPWR-class	<i>A Reference Class which contains the parameters of a PWR model.</i>
----------------	------------------------------------------------------------------------

---

### Description

ParamPWR contains all the parameters of a PWR model. The parameters are calculated by the initialization Method and then updated by the Method dynamic programming (here dynamic programming)

### Fields

- X Numeric vector of length  $m$  representing the covariates/inputs  $x_1, \dots, x_m$ .
- Y Numeric vector of length  $m$  representing the observed response/output  $y_1, \dots, y_m$ .
- m Numeric. Length of the response/output vector Y.
- K The number of regimes (PWR components).
- p The order of the polynomial regression. p is fixed to 3 by default.
- gamma Set of transition points. gamma is a column matrix of size  $(K + 1, 1)$ .
- beta Parameters of the polynomial regressions. beta is a matrix of dimension  $(p + 1, K)$ , with p the order of the polynomial regression. p is fixed to 3 by default.
- sigma2 The variances for the K regimes. sigma2 is a matrix of size  $(K, 1)$ .
- phi A list giving the regression design matrices for the polynomial and the logistic regressions.

### Methods

- computeDynamicProgram(C1, K) Method which implements the dynamic programming based on the cost matrix C1 and the number of regimes/segments K.
- computeParam() Method which estimates the parameters beta and sigma2 knowing the transition points gamma.

---

ParamRHLP-class	<i>A Reference Class which contains parameters of a RHLP model.</i>
-----------------	---------------------------------------------------------------------

---

### Description

ParamRHLP contains all the parameters of a RHLP model. The parameters are calculated by the initialization Method and then updated by the Method implementing the M-Step of the EM algorithm.

### Fields

- X Numeric vector of length  $m$  representing the covariates/inputs  $x_1, \dots, x_m$ .
- Y Numeric vector of length  $m$  representing the observed response/output  $y_1, \dots, y_m$ .
- m Numeric. Length of the response/output vector Y.
- K The number of regimes (RHLP components).
- p The order of the polynomial regression.
- q The dimension of the logistic regression. For the purpose of segmentation, it must be set to 1.

- variance\_type Character indicating if the model is homoskedastic (variance\_type = "homoskedastic") or heteroskedastic (variance\_type = "heteroskedastic"). By default the model is heteroskedastic.
- W Parameters of the logistic process.  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_{K-1})$  is a matrix of dimension  $(q+1, K-1)$ , with q the order of the logistic regression. q is fixed to 1 by default.
- beta Parameters of the polynomial regressions.  $\beta = (\beta_1, \dots, \beta_K)$  is a matrix of dimension  $(p+1, K)$ , with p the order of the polynomial regression. p is fixed to 3 by default.
- sigma2 The variances for the K regimes. If RHLP model is heteroskedastic (variance\_type = "heteroskedastic") then sigma2 is a matrix of size  $(K, 1)$  (otherwise RHLP model is homoskedastic (variance\_type = "homoskedastic") and sigma2 is a matrix of size  $(1, 1)$ ).
- nu The degree of freedom of the RHLP model representing the complexity of the model.
- phi A list giving the regression design matrices for the polynomial and the logistic regressions.

## Methods

- initParam(try\_algo = 1) Method to initialize parameters W, beta and sigma2.  
If try\_algo = 1 then beta and sigma2 are initialized by segmenting the time series Y uniformly into K contiguous segments. Otherwise, W, beta and sigma2 are initialized by segmenting randomly the time series Y into K segments.
- MStep(statRHLP, verbose\_IRLS) Method which implements the M-step of the EM algorithm to learn the parameters of the RHLP model based on statistics provided by the object statRHLP of class [StatRHLP](#) (which contains the E-step).

---

selectHMMR	<i>selectHMMR implements a model selection procedure to select an optimal HMMR model with unknown structure.</i>
------------	------------------------------------------------------------------------------------------------------------------

---

## Description

selectHMMR implements a model selection procedure to select an optimal HMMR model with unknown structure.

## Usage

```
selectHMMR(X, Y, Kmin = 1, Kmax = 10, pmin = 0, pmax = 4,
  criterion = c("BIC", "AIC"))
```

## Arguments

X	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
Y	Numeric vector of length $m$ representing the observed response/output $y_1, \dots, y_m$ .
Kmin	The minimum number of regimes (HMMR components).
Kmax	The maximum number of regimes (HMMR components).
pmin	The minimum order of the polynomial regression.
pmax	The maximum order of the polynomial regression.
criterion	The criterion used to select the HMMR model ("BIC", "AIC").

## Details

selectHMMR selects the optimal HMMR model among a set of model candidates by optimizing a model selection criteria, including the Bayesian Information Criterion (BIC). This function first fits the different HMMR model candidates by varying the number of regimes  $K$  from  $K_{\min}$  to  $K_{\max}$  and the order of the polynomial regression  $p$  from  $p_{\min}$  to  $p_{\max}$ . The model having the highest value of the chosen selection criterion is then selected.

## Value

selectHMMR returns an object of class [ModelHMMR](#) representing the selected HMMR model according to the chosen criterion.

## See Also

[ModelHMMR](#)

---

selectMHMMR	<i>selectMHMMR implements a model selection procedure to select an optimal MHMMR model with unknown structure.</i>
-------------	--------------------------------------------------------------------------------------------------------------------

---

## Description

selectMHMMR implements a model selection procedure to select an optimal MHMMR model with unknown structure.

## Usage

```
selectMHMMR(X, Y, Kmin = 1, Kmax = 10, pmin = 0, pmax = 4,
  criterion = c("BIC", "AIC"))
```

## Arguments

$X$	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
$Y$	Matrix of size $(m, d)$ representing a $d$ dimension time series observed at points $1, \dots, m$ .
$K_{\min}$	The minimum number of regimes (c components).
$K_{\max}$	The maximum number of regimes (MHMMR components).
$p_{\min}$	The minimum order of the polynomial regression.
$p_{\max}$	The maximum order of the polynomial regression.
criterion	The criterion used to select the MHMMR model ("BIC", "AIC").

## Details

selectMHMMR selects the optimal MHMMR model among a set of model candidates by optimizing a model selection criteria, including the Bayesian Information Criterion (BIC). This function first fits the different MHMMR model candidates by varying the number of regimes  $K$  from  $K_{\min}$  to  $K_{\max}$  and the order of the polynomial regression  $p$  from  $p_{\min}$  to  $p_{\max}$ . The model having the highest value of the chosen selection criterion is then selected.

**Value**

selectMHMMR returns an object of class [ModelMHMMR](#) representing the selected MHMMR model according to the chosen criterion.

**See Also**

[ModelMHMMR](#)

---

selectMRHLP	<i>selecMRHLP implements a model selection procedure to select an optimal MRHLP model with unknown structure.</i>
-------------	-------------------------------------------------------------------------------------------------------------------

---

**Description**

selectMRHLP implements a model selection procedure to select an optimal MRHLP model with unknown structure.

**Usage**

```
selectMRHLP(X, Y, Kmin = 1, Kmax = 10, pmin = 0, pmax = 4,
  criterion = c("BIC", "AIC"))
```

**Arguments**

X	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
Y	Matrix of size $(m, d)$ representing a $d$ dimension function of X observed at points $1, \dots, m$ . Y is the observed response/output.
Kmin	The minimum number of regimes (MRHLP components).
Kmax	The maximum number of regimes (MRHLP components).
pmin	The minimum order of the polynomial regression.
pmax	The maximum order of the polynomial regression.
criterion	The criterion used to select the MRHLP model ("BIC", "AIC").

**Details**

selectMRHLP selects the optimal MRHLP model among a set of model candidates by optimizing a model selection criteria, including the Bayesian Information Criterion (BIC). This function first fits the different MRHLP model candidates by varying the number of regimes K from Kmin to Kmax and the order of the polynomial regression p from pmin to pmax. The model having the highest value of the chosen selection criterion is then selected.

**Value**

selectMRHLP returns an object of class [ModelMRHLP](#) representing the selected MRHLP model according to the chosen criterion.

**See Also**

[ModelMRHLP](#)

---

selectRHLP	<i>selecRHLP implements a model selection procedure to select an optimal RHLP model with unknown structure.</i>
------------	-----------------------------------------------------------------------------------------------------------------

---

## Description

selectRHLP implements a model selection procedure to select an optimal RHLP model with unknown structure.

## Usage

```
selectRHLP(X, Y, Kmin = 1, Kmax = 10, pmin = 0, pmax = 4,
  criterion = c("BIC", "AIC"))
```

## Arguments

X	Numeric vector of length $m$ representing the covariates/inputs $x_1, \dots, x_m$ .
Y	Numeric vector of length $m$ representing the observed response/output $y_1, \dots, y_m$ .
Kmin	The minimum number of regimes (RHLP components).
Kmax	The maximum number of regimes (RHLP components).
pmin	The minimum order of the polynomial regression.
pmax	The maximum order of the polynomial regression.
criterion	The criterion used to select the RHLP model ("BIC", "AIC").

## Details

selectRHLP selects the optimal MRHLP model among a set of model candidates by optimizing a model selection criteria, including the Bayesian Information Criterion (BIC). This function first fits the different RHLP model candidates by varying the number of regimes  $K$  from  $K_{\min}$  to  $K_{\max}$  and the order of the polynomial regression  $p$  from  $p_{\min}$  to  $p_{\max}$ . The model having the highest value of the chosen selection criterion is then selected.

## Value

selectRHLP returns an object of class [ModelRHLP](#) representing the selected RHLP model according to the chosen criterion.

## See Also

[ModelRHLP](#)

StatHMMR-class

*A Reference Class which contains statistics of a HMMR model.***Description**

StatHMMR contains all the statistics associated to a [HMMR](#) model. It mainly includes the E-Step of the EM algorithm calculating the posterior distribution of the hidden variables (ie the smoothing probabilities), as well as the calculation of the prediction and filtering probabilities, the log-likelihood at each step of the algorithm and the obtained values of model selection criteria..

**Fields**

- tau\_tk** Matrix of size  $(m, K)$  giving the posterior probability that the observation  $Y_i$  originates from the  $k$ -th regression model.
- alpha\_tk** Matrix of size  $(m, K)$  giving the forwards probabilities:  $P(Y_1, \dots, Y_t, z_t = k)$ .
- beta\_tk** Matrix of size  $(m, K)$ , giving the backwards probabilities:  $P(Y_{t+1}, \dots, Y_m | z_t = k)$ .
- xi\_tkl** Array of size  $(m - 1, K, K)$  giving the joint post probabilities:  $xi_{tk}[t, k, l] = P(z_t = k, z_{t-1} = l | \mathbf{Y})$  for  $t = 2, \dots, m$ .
- f\_tk** Matrix of size  $(m, K)$  giving the cumulative distribution function  $f(y_t | z_t = k)$ .
- log\_f\_tk** Matrix of size  $(m, K)$  giving the logarithm of the cumulative distribution **f\_tk**.
- loglik** Numeric. Log-likelihood of the HMMR model.
- stored\_loglik** Numeric vector. Stored values of the log-likelihood at each iteration of the EM algorithm.
- klas** Column matrix of the labels issued from **z\_ik**. Its elements are  $klas(i) = k, k = 1, \dots, K$ .
- z\_ik** Hard segmentation logical matrix of dimension  $(m, K)$  obtained by the Maximum a posteriori (MAP) rule:  $z_{ik} = 1$  if  $z_{ik} = \arg \max_s P(z_i = s | \mathbf{Y}) = tau_{tk}$ ; 0 otherwise,  $k = 1, \dots, K$ .
- state\_probs** Matrix of size  $(m, K)$  giving the distribution of the Markov chain.  $P(z_1, \dots, z_m; \pi, \mathbf{A})$  with  $\pi$  the prior probabilities (field **prior** of the class [ParamHMMR](#)) and  $\mathbf{A}$  the transition matrix (field **trans\_mat** of the class [ParamHMMR](#)) of the Markov chain.
- BIC** Numeric. Value of BIC (Bayesian Information Criterion).
- AIC** Numeric. Value of AIC (Akaike Information Criterion).
- regressors** Matrix of size  $(m, K)$  giving the values of the estimated polynomial regression components.
- predict\_prob** Matrix of size  $(m, K)$  giving the prediction probabilities:  $P(z_t = k | y_1, \dots, y_{t-1})$ .
- predicted** Row matrix of size  $(m, 1)$  giving the sum of the polynomial components weighted by the prediction probabilities **predict\_prob**.
- filter\_prob** Matrix of size  $(m, K)$  giving the filtering probabilities  $Pr(z_t = k | y_1, \dots, y_t)$ .
- filtered** Row matrix of size  $(m, 1)$  giving the sum of the polynomial components weighted by the filtering probabilities.
- smoothed\_regressors** Matrix of size  $(m, K)$  giving the polynomial components weighted by the posterior probability **tau\_tk**.
- smoothed** Row matrix of size  $(m, 1)$  giving the sum of the polynomial components weighted by the posterior probability **tau\_tk**.



## Methods

`computeLikelihood(paramHMMR)` Method to compute the log-likelihood based on some parameters given by the object `paramHMMR` of class [ParamHMMR](#).

`computeStats(paramHMMR)` Method used in the EM algorithm to compute statistics based on parameters provided by the object `paramHMMR` of class [ParamHMMR](#).

`EStep(paramHMMR)` Method used in the EM algorithm to update statistics based on parameters provided by the object `paramHMMR` of class [ParamHMMR](#) (prior and posterior probabilities).

`MAP()` MAP calculates values of the fields `z_ik` and `klas` by applying the Maximum A Posteriori Bayes allocation rule.

$$z_{ik} = 1 \text{ if } z_{ik} = \arg \max_s P(z_i = s | \mathbf{Y}) = \text{tau\_tk}; 0 \text{ otherwise}$$

## See Also

[ParamHMMR](#)

---

StatMHMMR-class	<i>A Reference Class which contains statistics of a MHMMR model.</i>
-----------------	----------------------------------------------------------------------

---

## Description

StatMHMMR contains all the statistics associated to a [MHMMR](#) model. It mainly includes the E-Step of the EM algorithm calculating the posterior distribution of the hidden variables (ie the smoothing probabilities), as well as the calculation of the prediction and filtering probabilities, the log-likelihood at each step of the algorithm and the obtained values of model selection criteria..

## Fields

`tau_tk` Matrix of size  $(m, K)$  giving the posterior probability that the observation  $Y_i$  originates from the  $k$ -th regression model.

`alpha_tk` Matrix of size  $(m, K)$  giving the forwards probabilities:  $P(Y_1, \dots, Y_t, z_t = k)$ .

`beta_tk` Matrix of size  $(m, K)$ , giving the backwards probabilities:  $P(Y_{t+1}, \dots, Y_m | z_t = k)$ .

`xi_tkl` Array of size  $(m - 1, K, K)$  giving the joint post probabilities:  $xi_{tk}[t, k, l] = P(z_t = k, z_{t-1} = l | \mathbf{Y})$  for  $t = 2, \dots, m$ .

`f_tk` Matrix of size  $(m, K)$  giving the cumulative distribution function  $f(y_t | z_t = k)$ .

`log_f_tk` Matrix of size  $(m, K)$  giving the logarithm of the cumulative distribution `f_tk`.

`loglik` Numeric. Log-likelihood of the MHMMR model.

`stored_loglik` Numeric vector. Stored values of the log-likelihood at each iteration of the EM algorithm.

`klas` Column matrix of the labels issued from `z_ik`. Its elements are  $klas(i) = k, k = 1, \dots, K$ .

`z_ik` Hard segmentation logical matrix of dimension  $(m, K)$  obtained by the Maximum a posteriori (MAP) rule:  $z_{ik} = 1$  if  $z_{ik} = \arg \max_s P(z_i = s | \mathbf{Y}) = \text{tau\_tk}$ ; 0 otherwise,  $k = 1, \dots, K$ .

`state_probs` Matrix of size  $(m, K)$  giving the distribution of the Markov chain.  $P(z_1, \dots, z_m; \pi, \mathbf{A})$  with  $\pi$  the prior probabilities (field `prior` of the class [ParamMHMMR](#)) and  $\mathbf{A}$  the transition matrix (field `trans_mat` of the class [ParamMHMMR](#)) of the Markov chain.

`BIC` Numeric. Value of BIC (Bayesian Information Criterion).

AIC Numeric. Value of AIC (Akaike Information Criterion).

regressors Matrix of size  $(m, K)$  giving the values of the estimated polynomial regression components.

predict\_prob Matrix of size  $(m, K)$  giving the prediction probabilities:  $P(z_t = k | y_1, \dots, y_{t-1})$ .

predicted Row matrix of size  $(m, 1)$  giving the sum of the polynomial components weighted by the prediction probabilities predict\_prob.

filter\_prob Matrix of size  $(m, K)$  giving the filtering probabilities  $Pr(z_t = k | y_1, \dots, y_t)$ .

filtered Row matrix of size  $(m, 1)$  giving the sum of the polynomial components weighted by the filtering probabilities.

smoothed\_regressors Matrix of size  $(m, K)$  giving the polynomial components weighted by the posterior probability tau\_tk.

smoothed Row matrix of size  $(m, 1)$  giving the sum of the polynomial components weighted by the posterior probability tau\_tk.

## Methods

computeLikelihood(paramMHMMR) Method to compute the log-likelihood based on some parameters given by the object paramMHMMR of class [ParamMHMMR](#).

computeStats(paramMHMMR) Method used in the EM algorithm to compute statistics based on parameters provided by the object paramMHMMR of class [ParamMHMMR](#).

EStep(paramMHMMR) Method used in the EM algorithm to update statistics based on parameters provided by the object paramMHMMR of class [ParamMHMMR](#) (prior and posterior probabilities).

MAP() MAP calculates values of the fields z\_ik and klas by applying the Maximum A Posteriori Bayes allocation rule.

$z_{ik} = 1$  if  $z_{ik} = \arg \max_s P(z_i = s | \mathbf{Y}) = \text{tau\_tk}$ ; 0 otherwise

## See Also

[ParamMHMMR](#)

---

StatMRHLP-class

*A Reference Class which contains statistics of a MRHLP model.*

---

## Description

StatMRHLP contains all the statistics associated to a [MRHLP](#) model. It mainly includes the E-Step of the EM algorithm calculating the posterior distribution of the hidden variables, as well as the calculation of the log-likelihood at each step of the algorithm and the obtained values of model selection criteria..

## Fields

pi\_ik Matrix of size  $(m, K)$  representing the prior/logistic probabilities  $\pi_k(x_i; \Psi) = P(z_i = k | x; \Psi)$  of the latent variable  $z_i, i = 1, \dots, m$ .

z\_ik Hard segmentation logical matrix of dimension  $(m, K)$  obtained by the Maximum a posteriori (MAP) rule:  $z_{ik} = 1$  if  $z_{ik} = \arg \max_s \pi_s(x_i; \Psi)$ ; 0 otherwise,  $k = 1, \dots, K$ .

klas Column matrix of the labels issued from z\_ik. Its elements are  $klas(i) = k, k = 1, \dots, K$ .

`tau_ik` Matrix of size  $(m, K)$  giving the posterior probability that the observation  $Y_i$  originates from the  $k$ -th regression model.

`polynomials` Array of size  $(m, d, K)$  giving the values of the estimated polynomial regression components.

`weighted_polynomials` Array of size  $(m, d, K)$  giving the values of the estimated polynomial regression components weighted by the prior probabilities `pi_ik`.

`Ex` Matrix of size  $(m, d)$ . `Ex` is the curve expectation (estimated signal): sum of the polynomial components weighted by the logistic probabilities `pi_ik`.

`loglik` Numeric. Observed-data log-likelihood of the MRHLP model.

`com_loglik` Numeric. Complete-data log-likelihood of the MRHLP model.

`stored_loglik` Numeric vector. Stored values of the log-likelihood at each EM iteration.

`stored_com_loglik` Numeric vector. Stored values of the Complete log-likelihood at each EM iteration.

`BIC` Numeric. Value of BIC (Bayesian Information Criterion).

`ICL` Numeric. Value of ICL (Integrated Completed Likelihood).

`AIC` Numeric. Value of AIC (Akaike Information Criterion).

`log_piik_fik` Matrix of size  $(m, K)$  giving the values of the logarithm of the joint probability  $P(y_i, z_i = k | \mathbf{x}, \Psi)$ ,  $i = 1, \dots, m$ .

`log_sum_piik_fik` Column matrix of size  $m$  giving the values of  $\log \sum_{k=1}^K P(y_i, z_i = k | \mathbf{x}, \Psi)$ ,  $i = 1, \dots, m$ .

## Methods

`computeLikelihood(reg_irls)` Method to compute the log-likelihood. `reg_irls` is the value of the regularization part in the IRLS algorithm.

`computeStats(paramMRHLP)` Method used in the EM algorithm to compute statistics based on parameters provided by the object `paramMRHLP` of class [ParamMRHLP](#).

`EStep(paramMRHLP)` Method used in the EM algorithm to update statistics based on parameters provided by the object `paramMRHLP` of class [ParamMRHLP](#) (prior and posterior probabilities).

`MAP()` `MAP` calculates values of the fields `z_ik` and `klas` by applying the Maximum A Posteriori Bayes allocation rule.

$$z_{ik} = 1 \text{ if } k = \arg \max_s \pi_s(x_i; \Psi); 0 \text{ otherwise}$$

## See Also

[ParamMRHLP](#)

---

StatPWR-class

*A Reference Class which contains statistics of a PWR model.*

---

## Description

StatPWR contains all the statistics associated to a [PWR](#) model.

**Fields**

- z\_ik** Logical matrix of dimension  $(m, K)$  giving the class vector.
- klas** Column matrix of the labels issued from **z\_ik**. Its elements are  $klas(i) = k, k = 1, \dots, K$ .
- mean\_function** Approximation of the time series given the estimated parameters. **mean\_function** is a matrix of size  $(m, 1)$ .
- regressors** Matrix of size  $(m, K)$  giving the values of the estimated polynomial regression components.
- objective** Numeric. Value of the objective function.

**Methods**

- computeStats(paramPWR)** Method used at the end of the dynamic programming algorithm to compute statistics based on parameters provided by **paramPWR**.

**See Also**

[ParamPWR](#)

---

StatRHLP-class

*A Reference Class which contains statistics of a RHLP model.*

---

**Description**

StatRHLP contains all the statistics associated to a [RHLP](#) model. It mainly includes the E-Step of the EM algorithm calculating the posterior distribution of the hidden variables, as well as the calculation of the log-likelihood at each step of the algorithm and the obtained values of model selection criteria..

**Fields**

- pi\_ik** Matrix of size  $(m, K)$  representing the prior/logistic probabilities  $\pi_k(x_i; \Psi) = P(z_i = k | x; \Psi)$  of the latent variable  $z_i, i = 1, \dots, m$ .
- z\_ik** Hard segmentation logical matrix of dimension  $(m, K)$  obtained by the Maximum a posteriori (MAP) rule:  $z_{ik} = 1$  if  $z_{ik} = \arg \max_s \pi_s(x_i; \Psi)$ ; 0 otherwise,  $k = 1, \dots, K$ .
- klas** Column matrix of the labels issued from **z\_ik**. Its elements are  $klas(i) = k, k = 1, \dots, K$ .
- tau\_ik** Matrix of size  $(m, K)$  giving the posterior probability that the observation  $Y_i$  originates from the  $k$ -th regression model.
- polynomials** Matrix of size  $(m, K)$  giving the values of the estimated polynomial regression components.
- Ex** Column matrix of dimension  $m$ . **Ex** is the curve expectation (estimated signal): sum of the polynomial components weighted by the logistic probabilities **pi\_ik**.
- loglik** Numeric. Observed-data log-likelihood of the RHLP model.
- com\_loglik** Numeric. Complete-data log-likelihood of the RHLP model.
- stored\_loglik** Numeric vector. Stored values of the log-likelihood at each EM iteration.
- stored\_com\_loglik** Numeric vector. Stored values of the Complete log-likelihood at each EM iteration.

BIC Numeric. Value of BIC (Bayesian Information Criterion).

ICL Numeric. Value of ICL (Integrated Completed Likelihood).

AIC Numeric. Value of AIC (Akaike Information Criterion).

log\_piik\_fik Matrix of size  $(m, K)$  giving the values of the logarithm of the joint probability  $P(y_i, z_i = k | \mathbf{x}, \Psi), i = 1, \dots, m$ .

log\_sum\_piik\_fik Column matrix of size  $m$  giving the values of  $\log \sum_{k=1}^K P(y_i, z_i = k | \mathbf{x}, \Psi), i = 1, \dots, m$ .

## Methods

computeLikelihood(reg\_irls) Method to compute the log-likelihood. reg\_irls is the value of the regularization part in the IRLS algorithm.

computeStats(paramRHLP) Method used in the EM algorithm to compute statistics based on parameters provided by the object paramRHLP of class [ParamRHLP](#).

EStep(paramRHLP) Method used in the EM algorithm to update statistics based on parameters provided by the object paramRHLP of class [ParamRHLP](#) (prior and posterior probabilities).

MAP() MAP calculates values of the fields z\_ik and klas by applying the Maximum A Posteriori Bayes allocation rule.

$$z_{ik} = 1 \text{ if } k = \arg \max_s \pi_s(x_i; \Psi); 0 \text{ otherwise}$$

## See Also

[ParamRHLP](#)

---

univrealdataset	<i>Time series representing the electrical power consumption during a railway switch operation</i>
-----------------	----------------------------------------------------------------------------------------------------

---

## Description

This dataset is provided for illustration only; It is issued from the switch railway monitoring domain. The switch mechanism enables trains to be guided from one track to another at a railway junction. During each switch operation, a set of measurements are recorded. Each measurement represents the consumed electrical power. The resulting time series present regime changes.

## Usage

```
univrealdataset
```

## Format

A data frame with 562 rows and 3 columns:

**x** The covariate variables which are the sampling time in this time-series case.

**y1** Measurements of the electrical power consumed during time for a first example of switch operations.

**y2** Measurements of the electrical power consumed during during time for another example of switch operations.

---

univtoydataset	<i>A simulated non-stationary time series with regime changes.</i>
----------------	--------------------------------------------------------------------

---

**Description**

A simulated non-stationary time series with regime changes. This time series is used for illustration.

**Usage**

```
univtoydataset
```

**Format**

A data frame with 670 rows and 2 columns:

**x** The covariate variable which is the time in this time-series case.

**y** The time series. The latter has been generated as follows:

- First regime: 100 values of standard Normally distributed random numbers.
- Second regime: 120 values of Normally distributed random numbers with mean 7 and unit variance.
- Third regime: 200 values of Normally distributed random numbers with mean 4 and unit variance.
- Fourth regime: 100 values of Normally distributed random numbers with mean -2 and unit variance.
- Fifth regime: 150 values of Normally distributed random numbers with mean 3.5 and unit variance.

# Index

## \*Topic **datasets**

- multivrealdataset, [14](#)
- multivtoydataset, [15](#)
- univrealdataset, [29](#)
- univtoydataset, [30](#)

- emHMMR, [4](#)
- emMHMMR, [5](#)
- emMRHLP, [6](#)
- emRHLP, [7](#)

- fitPWRFisher, [8](#)

- hmmProcess, [9](#)
- HMMR, [24](#)

- MData, [17](#), [18](#)
- MData (MData-class), [10](#)
- MData-class, [10](#)
- MHMMR, [25](#)
- mkStochastic, [10](#)
- ModelHMMR, [5](#), [21](#)
- ModelHMMR (ModelHMMR-class), [11](#)
- ModelHMMR-class, [11](#)
- ModelMHMMR, [6](#), [22](#)
- ModelMHMMR (ModelMHMMR-class), [11](#)
- ModelMHMMR-class, [11](#)
- ModelMRHLP, [7](#), [22](#)
- ModelMRHLP (ModelMRHLP-class), [12](#)
- ModelMRHLP-class, [12](#)
- ModelPWR, [9](#)
- ModelPWR (ModelPWR-class), [13](#)
- ModelPWR-class, [13](#)
- ModelRHLP, [8](#), [23](#)
- ModelRHLP (ModelRHLP-class), [13](#)
- ModelRHLP-class, [13](#)
- MRHLP, [26](#)
- multivrealdataset, [14](#)
- multivtoydataset, [15](#)

- ParamHMMR, [5](#), [11](#), [24](#), [25](#)
- ParamHMMR (ParamHMMR-class), [16](#)
- ParamHMMR-class, [16](#)
- ParamMHMMR, [6](#), [11](#), [12](#), [25](#), [26](#)
- ParamMHMMR (ParamMHMMR-class), [17](#)

- ParamMHMMR-class, [17](#)
- ParamMRHLP, [7](#), [12](#), [13](#), [27](#)
- ParamMRHLP (ParamMRHLP-class), [18](#)
- ParamMRHLP-class, [18](#)
- ParamPWR, [9](#), [13](#), [28](#)
- ParamPWR (ParamPWR-class), [19](#)
- ParamPWR-class, [19](#)
- ParamRHLP, [8](#), [13](#), [14](#), [29](#)
- ParamRHLP (ParamRHLP-class), [19](#)
- ParamRHLP-class, [19](#)
- PWR, [27](#)

- RHLP, [28](#)

- samurais (samurais-package), [3](#)
- samurais-package, [3](#)
- selectHMMR, [20](#)
- selectMHMMR, [21](#)
- selectMRHLP, [22](#)
- selectRHLP, [23](#)
- StatHMMR, [5](#), [11](#), [16](#)
- StatHMMR (StatHMMR-class), [24](#)
- StatHMMR-class, [24](#)
- StatMHMMR, [6](#), [11](#), [12](#), [17](#)
- StatMHMMR (StatMHMMR-class), [25](#)
- StatMHMMR-class, [25](#)
- StatMRHLP, [7](#), [12](#), [13](#), [18](#)
- StatMRHLP (StatMRHLP-class), [26](#)
- StatMRHLP-class, [26](#)
- StatPWR, [9](#), [13](#)
- StatPWR (StatPWR-class), [27](#)
- StatPWR-class, [27](#)
- StatRHLP, [8](#), [13](#), [14](#), [20](#)
- StatRHLP (StatRHLP-class), [28](#)
- StatRHLP-class, [28](#)

- univrealdataset, [29](#)
- univtoydataset, [30](#)