Package 'MPWR'

August 6, 2019

Type Package

Title Multiple Polynomial Piecewise Regression ('MPWR') for Optimal Multivariate Time-Series Segmentation

Version 0.1.0

Description A multiple polynomial piecewise regression model for the optimal segmentation of multivariate time series with regime changes. It uses MLE for the estimation of the regression parameters with dynamic programming for the segmentation. Originally developped and written in 'Matlab' by Faicel Chamroukhi <https://github.com/fchamroukhi/MPWR>

URL https://github.com/fchamroukhi/MPWR

License GPL (>= 3)

Depends R (>= 2.10)

Imports methods,
stats,
Rcpp

Suggests knitr,
rmarkdown

LinkingTo Rcpp,
RcppArmadillo

Collate MPWR-package.R
RcppExports.R
utils.R
MData.R
dynamicProg.R
ParamMPWR.R
StatMPWR.R
ModelMPWR.R
fitMPWRFisher.R
data-realdataset.R
data-toydataset.R

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 6.1.1

R topics documented:

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

Description

fitMPWRFisher is used to fit a Mulitvariate 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

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

Arguments

`X`	Numeric vector of length *m* representing the covariates/inputs $x_1, \ldots, x_m$.
`Y`	Matrix of size (m, d) representing a d dimension function of `X` observed at points $1, \ldots, m$. `Y` is the observed response/output.
`K`	The number of regimes/segments (PWR components).
`p`	Optional. The order of the polynomial regression. By default, `p` is set at 3.

Details

fitMPWRFisher function implements an optimized dynamic programming algorithm of the MPWR 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 ParamMPWR).

Value

fitMPWRFisher returns an object of class ModelMPWR.

See Also

ModelMPWR, ParamMPWR, StatMPWR

Examples

```
data(toydataset)
x <- toydataset$x
Y <- as.matrix(toydataset[,c("y1", "y2", "y3")])

mpwr <- fitMPWRFisher(X = x, Y = Y, K = 5, p = 1)

mpwr$summary()

mpwr$plot()
```

`MData-class` *A Reference Class which represents multivariate data.*

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 $1, \ldots, m$.

`ModelMPWR-class` *A Reference Class which represents a fitted MPWR model.*

Description

ModelMPWR represents an estimated MPWR model.

Fields

`param` A ParamMPWR object. It contains the estimated values of the parameters.

`stat` A StatMPWR object. It contains all the statistics associated to the MPWR model.

Methods

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

`what` The type of graph requested:

- `"regressors"` = Polynomial regression components (field `regressors` of class StatMPWR).
- `"segmentation"` = Estimated signal (field `mean_function` of class StatMPWR).

`...` 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

ParamMPWR, StatMPWR

Examples

```
data(toydataset)
x <- toydataset$x
Y <- as.matrix(toydataset[,c("y1", "y2", "y3")])

mpwr <- fitMPWRFisher(X = x, Y = Y, K = 5, p = 1)

# mpwr is a ModelMPWR object. It contains some methods such as 'summary' and 'plot'
mpwr$summary()
mpwr$plot()

# mpwr has also two fields, stat and param which are reference classes as well

# Value of the objective function:
mpwr$stat$objective

# Parameters of the polynomial regressions:
mpwr$param$beta
```

ParamMPWR-class *A Reference Class which contains the parameters of a PWR model.*

Description

ParamMPWR 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 $x_1, \ldots, x_m$.

`Y` Numeric vector of length *m* representing the observed response/output $y_1, \ldots, y_m$.

`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 an array of dimension $(p + 1, d, K)$, with `p` the order of the polynomial regression, `d` the dimension of the multivariate time-series. `p` is fixed to 3 by default.

`sigma2` The variances for the `K` regimes. `sigma2` is an array of size (d, d, K).

`phi` A matrix giving the regression design matrix for the polynomial regression.

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`.

realdataset — *Time series representing the three acceleration components recorded over time with body mounted accelerometers during the activity of a given person.*

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

```
realdataset
```

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.

StatMPWR-class — *A Reference Class which contains statistics of a PWR model.*

Description

StatMPWR contains all the statistics associated to a MPWR 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 $klas(i) = k$, $k = 1, \ldots, K$.

`mean_function` Approximation of the time series given the estimated parameters. `mean_function` is a matrix of size (m, d).

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

`objective` Numeric. Value of the objective function.

Methods

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

See Also

ParamMPWR

`toydataset` *A simulated non-stationary multidimensional time series with regime changes.*

Description

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

Usage

```
toydataset
```

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.

Index