Package 'MMLR'

February 3, 2019

Type Package

Title Fitting Markov-Modulated Linear Regression Models

Version 0.1.0

Maintainer Nadezda Spiridovska <Spiridovska.N@tsi.lv>

Description A set of tools for fitting Markov-modulated linear regression, where responses Y(t) are time-additive, and model operates in the external environment, which is described as a continuous time Markov chain with finite state space.
Model is proposed by Alexander Andronov (2012) <arXiv:1901.09600v1> and algorithm of parameters estimation is based on eigenvalues and eigenvectors decomposition.
Also, package will provide a set of data simulation tools for Markov-modulated linear regression (for academical/research purposes).
Research project No. 1.1.1.2/VIAA/1/16/075.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports matlib

RoxygenNote 6.1.1

NeedsCompilation no

Author Nadezda Spiridovska [aut, cre],
Diana Santalova [ctb]

Repository CRAN

Date/Publication 2019-02-03 16:33:17 UTC

R topics documented:

`Aver_soj_time` *Calculating average sojourn time in each state*

Description

Calculating expectation of sojourn times in states (currently two states) for the observed time and for given initial state, using eigenvalues and eigenvectors.

Usage

```
Aver_soj_time(ii, tau_observed, Q)
```

Arguments

`ii`	number (scalar), currently 1 or 2
`tau_observed`	number (scalar), observed time
`Q`	Matrix (m x m, currently m = 2, see Details

Details

Calculating expectation of sojourn times in states 1 and 2 for the observed time (tau_observed) and if initial state is given (ii). Matrix Q is so-called Generator matrix: Q = lambda – Lambda, where lambda is matrix with transition rates from state s_i to state s_j, and Lambda is diagonal matrix with a vector (Lambda_1, . . . , Lambda_m) on the main diagonal, where m is a number of states of external environment (currently m = 2). Eigenvalues and eigenvectors are used in calculations.

Value

Vector of average sojourn times in each state. Vector components in total should give observation time (tau_observed).

Examples

```
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
m <- nrow(lambda)
ld <- as.matrix(rowSums(lambda))
Lambda <- diag(as.vector(ld))
Generator <- t(lambda) - Lambda
Aver_soj_time(1,10,Generator)
Aver_soj_time(2,5,Generator)
```

B_est2 *Estimantion of unknown Markov-modulated linear regression model parameters using GLSM*

Description

This function is used to fit Markov-modulated linear regression models with two states of external environment. This function estimates Markov-modulated linear regression model parameters, using GLSM. Function uses the algorithm based on eigenvalues and eigenvectors decompositions.

Usage

```
B_est2(tGiven, initState, X, Y, lambda, W = TRUE)
```

Arguments

tGiven	Vector n, n – number of observations
initState	Vector n, n – number of observations
X	Matrix (n x k+1), k – number of regressors (plus intercept)
Y	Vector n, n – number of observations
lambda	Matrix (m x m), m – number of states
W	an optional logical variable indicating should vector of weights be used in the fitting process or not. If TRUE, matrix with weights is used (that is, inverse values to tGiven – observed times).

Details

Function calculates the following expression: addImage(doc, "vecB.png") where vector of average sojourn times in each state t_i is calculated using function Aver_soj_time, t_i is an element of tGiven, x_i is a vector of matrix X.

Value

Vector of estimated parameters

Examples

```
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
X <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,9,1,2,3,2,3,5,2,2))
tGiven <- matrix (c(6,4.8,1,2.6,6.4,1.7,2.9,4.4,1.5,3.4), nrow = 10, ncol = 1)
Y <- matrix(c (5.7, 9.9, 7.8, 14.5, 8.2, 14.5, 32.2, 3.8, 16.5, 7.7),nrow = 10, ncol = 1)
initState <- matrix (c(2,1,1,2,2,2,1,1,2,1),nrow = 10, ncol = 1)
B_est2(tGiven,initState,X,Y,lambda,W = 1)
```

Xreg *Preparing data for parameter estimation procedure*

Description

Regressors matrix formation taking into account observation times and initial states. Kronecker product is used.

Usage

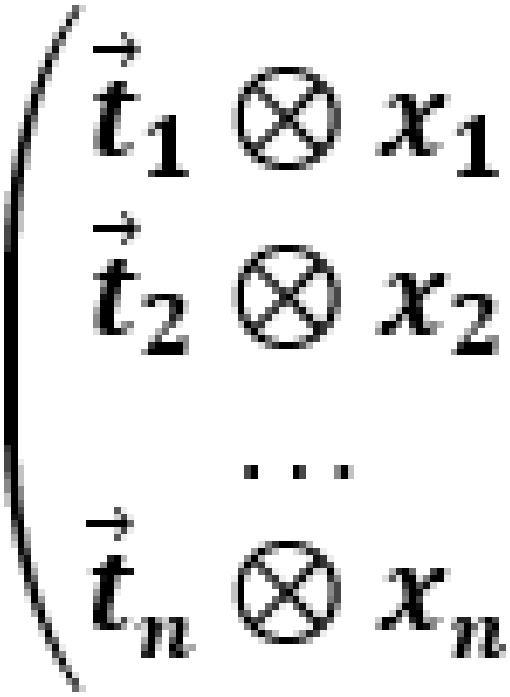
```
Xreg(tGiven, initState, X, lambda)
```

Arguments

tGiven	Vector n
initState	Vector n
X	Matrix (n x k+1), k – number of regressors (plus intercept)
lambda	Matrix (m x m), m – number of states

Details

Function calculates the following expression addImage(doc, "matrix.png") where vector of average



sojourn times in each state is calculated using function Aver_soj_time.

Value

Matrix (n x 2(k+1))

Examples

```
X <- cbind(rep_len(1,10),c(2,5,7,3,1,1,2,2,3,6), c(5,9,1,2,3,2,3,5,2,2))
tGiven <- matrix (c(6,4.8,1,2.6,6.4,1.7,2.9,4.4,1.5,3.4), nrow = 10, ncol = 1)
initState <- matrix (c(2,1,1,2,2,2,1,1,2,1),nrow = 10, ncol = 1)
lambda <- matrix(c(0, 0.33, 0.45, 0), nrow = 2, ncol = 2, byrow = TRUE)
Xreg(tGiven, initState, X, lambda)
```

Index