Package ‘FourWayHMM’

November 30, 2021

Title Parsimonious Hidden Markov Models for Four-Way Data

Version 1.0.0

Description Implements parsimonious hidden Markov models for four-way data via expectation-conditional maximization algorithm, as described in Tomarchio et al. (2020) <arXiv:2107.04330>. The matrix-variate normal distribution is used as emission distribution. For each hidden state, parsimony is reached via the eigen-decomposition of the covariance matrices of the emission distribution. This produces a family of 98 parsimonious hidden Markov models.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports withr, snow, doSNOW, foreach, mclust, tensor, tidyr, data.table, LaplacesDemon

Depends R (>= 2.10)

NeedsCompilation no

Author Salvatore D. Tomarchio [aut, cre],
Antonio Punzo [aut],
Antonello Maruotti [aut]

Maintainer Salvatore D. Tomarchio <daniele.tomarchio@unict.it>

Repository CRAN

Date/Publication 2021-11-30 21:20:02 UTC

R topics documented:

HMM.fit ................................................................. 2
HMM.init ............................................................... 3
simX ................................................................. 4

Index 5
**HMM.fit**

*Fitting for parsimonious hidden Markov models for four-way data*

**Description**

Fits, by using an ECM algorithm, parsimonious hidden Markov models to the given four-way data. Parallel computing is implemented and highly recommended for a faster model fitting. The Bayesian information criterion (BIC) is used to select the best fitting model.

**Usage**

```r
HMM.fit(
  X,
  k = 1:3,
  init.par = NULL,
  mod.row = "all",
  mod.col = "all",
  ncores = 1,
  verbose = FALSE,
  ret.all = FALSE
)
```

**Arguments**

- **X**
  - An array of dimension $p \times r \times n \times t$, where $p$ is the number of variables in the rows of each data matrix, $r$ is the number of variables in the columns of each data matrix, $n$ is the number of data observations and $t$ is the number of times.

- **k**
  - An integer or a vector indicating the number of states of the models.

- **init.par**
  - The initial values for starting the algorithms, as produced by the `HMM.init()` function.

- **mod.row**
  - A character vector indicating the parsimonious structure of the row covariance matrix. Possible values are: "EII", "VII", "EEI", "VEI", "EVI", "VVI", "EEE", "VEE", "EVE", "EEV", "VVE", "VEV", "EVV", "VVV" or "all". When "all" is used, all of the 14 row parsimonious structures are considered.

- **mod.col**
  - A character vector indicating the parsimonious structure of the column covariance matrix. Possible values are: "II", "EI", "VI", "EE", "VE", "EV", "VV", or "all". When "all" is used, all of the 7 column parsimonious structures are considered.

- **ncores**
  - A positive integer indicating the number of cores used for running in parallel.

- **verbose**
  - A logical indicating whether the running output should be displayed.

- **ret.all**
  - A logical indicating whether to report the results of all the models or only those of the best model according to the BIC.
**HMM.init**

**Value**

A list with the following elements:

- **all.models**: The results related to all the fitted models (only when ret.all = TRUE).
- **BicWin**: The best fitting model according to the BIC.
- **Summary**: A quick table showing summary results for the best fitting model according to the BIC.
- **c.time**: Provides information on the computational times required to fit all the models for each state.

**Examples**

```r
data(simX)
init <- HMM.init(X = simX, k = 2, nstartR = 1)
res <- HMM.fit(X = simX, k = 2, init.par = init, mod.row = "VII", mod.col = "EE")
```

**Description**

Runs the initialization of the ECM algorithm used for fitting the parsimonious hidden Markov models. Parallel computing is implemented and highly recommended for a faster calculation.

**Usage**

```r
HMM.init(X, k = 1:3, nstartR = 100, ncores = 1, verbose = FALSE)
```

**Arguments**

- **X**: An array of dimension $p \times r \times n \times t$, where $p$ is the number of variables in the rows of each data matrix, $r$ is the number of variables in the columns of each data matrix, $n$ is the number of data observations and $t$ is the number of times.
- **k**: An integer or a vector indicating the number of states of the models.
- **nstartR**: An integer specifying the number of random starts to be considered.
- **ncores**: A positive integer indicating the number of cores used for running in parallel.
- **verbose**: A logical indicating whether the running output should be displayed.

**Value**

- **init**: A list of objects to be used by the `HMM.fit()` function.

**Examples**

```r
data(simX)
init <- HMM.init(X = simX, k = 2, nstartR = 1)
```
simX

Simulated Data

Description
A simulated four-way dataset with 2 states and VII - EE covariance structure.

Usage
```r
data(simX)
```

Format
A four-way array having \( p = 2 \) variables in the rows, \( r = 3 \) variables in the columns, \( n = 50 \) data observations and \( t = 10 \) times.
Index

* datasets
  simX, 4

HMM.fit, 2
HMM.init, 3

simX, 4