Package ‘MatrixHMM’

July 21, 2025

Title Parsimonious Families of Hidden Markov Models for Matrix-Variate Longitudinal Data

Version 1.0.0

Description Implements three families of parsimonious hidden Markov models (HMMs) for matrix-variate longitudinal data using the Expectation-Conditional Maximization (ECM) algorithm. The package supports matrix-variate normal, t, and contaminated normal distributions as emission distributions. For each hidden state, parsimony is achieved through the eigen-decomposition of the covariance matrices associated with the emission distribution. This approach results in a comprehensive set of 98 parsimonious HMMs for each type of emission distribution. Atypical matrix detection is also supported, utilizing the fitted (heavy-tailed) models.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

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

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Salvatore D. Tomarchio [aut, cre]

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

Repository CRAN

Date/Publication 2024-08-28 08:00:06 UTC

Contents

`atp.MVCN` *Atypical Detection Points Using Matrix-Variate Contaminated Normal Hidden Markov Models*

Description

Detects atypical matrices via matrix-variate contaminated normal Hidden Markov Models.

Usage

```
atp.MVCN(Y, pgood, class)
```

Arguments

`Y`	An array with dimensions `p` x `r` x `num` x `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, `num` is the number of data observations, and `t` is the number of time points.
`pgood`	An array with dimensions `num` x `t` x `k` containing the estimated probability of being typical for each point, given the time and state.
`class`	An `num` x `t` matrix containing the state memberships.

Value

An `num` x `t` matrix containing, for each observation and time, a 0 if it that matrix is typical and 1 otherwise.

Examples

```
data(”simData2”)
Y <- simData2$Y
init <- Eigen.HMM_init(Y = Y, k = 2, density = ”MVCN”, mod.row = ”EEE”, mod.col = ”EE”, nstartR = 1)
fit <- Eigen.HMM_fit(Y = Y, init.par = init, nThreads = 1)
atp <- atp.MVCN(Y = Y,
                pgood = fit[[”results”]][[1]][[1]][[1]][[”pgood”]],
                class = fit[[”results”]][[1]][[1]][[1]][[”class”]])
which(atp==1)
which(simData2[[”atp.tr”]]==1)
```

`atp.MVT` *Atypical Detection Points Using Matrix-Variate t Hidden Markov Models*

Description

Detects atypical matrices via matrix-variate t Hidden Markov Models given a specified value of `epsilon`.

Usage

```
atp.MVT(Y, M, U, V, class, epsilon)
```

Arguments

`Y`	An array with dimensions `p x r x num x 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, `num` is the number of data observations, and `t` is the number of time points.
`M`	An array with dimensions `p x p x k`, where `k` is the number of states, containing the mean matrices.
`U`	An array with dimensions `p x p x k`, where `k` is the number of states, containing the row covariance (scale) matrices.
`V`	An array with dimensions `r x r x k`, where `k` is the number of states, containing the column covariance (scale) matrices.
`class`	An `num x t` matrix containing the state memberships.
`epsilon`	A numeric value specifying the selected percentile of the chi-squared distribution with `pr` degrees of freedom.

Value

An `num x t` matrix containing, for each observation and time, a 0 if it that matrix is typical and 1 otherwise.

Examples

```
data("simData2")
Y <- simData2$Y
init <- Eigen.HMM_init(Y = Y, k = 2, density = "MVT", mod.row = "EEE", mod.col = "EE", nstartR = 1)
fit <- Eigen.HMM_fit(Y = Y, init.par = init, nThreads = 1)
atp <- atp.MVT(Y = Y, M = fit[["results"]][[1]][[1]][[1]][["M"]],
              U = fit[["results"]][[1]][[1]][[1]][["U"]],
              V = fit[["results"]][[1]][[1]][[1]][["V"]],
              class = fit[["results"]][[1]][[1]][[1]][["class"]],
              epsilon = 0.99)
which(atp==1)
which(simData2[["atp.tr"]]==1)
```

`Eigen.HMM_fit` *Fitting Parsimonious Hidden Markov Models for Matrix-Variate Longitudinal Data*

Description

Fits parsimonious Hidden Markov Models for matrix-variate longitudinal data using ECM algorithms. The models are based on the matrix-variate normal, matrix-variate t, and matrix-variate contaminated normal distributions. Parallel computing is implemented and highly recommended for faster model fitting.

Usage

```
Eigen.HMM_fit(
  Y,
  init.par = NULL,
  tol = 0.001,
  maxit = 500,
  nThreads = 1,
  verbose = FALSE
)
```

Arguments

`Y`	An array with dimensions `p` x `r` x `num` x `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, `num` is the number of data observations, and `t` is the number of time points.
`init.par`	A list of initial values for starting the algorithms, as generated by the `Eigen.HMM_init()` function.
`tol`	A numeric value specifying the tolerance level for the ECM algorithms' convergence.
`maxit`	A numeric value specifying the maximum number of iterations for the ECM algorithms.
`nThreads`	A positive integer indicating the number of cores to use for parallel processing.
`verbose`	A logical value indicating whether to display the running output.

Value

A list containing the following elements:

`results`	A list of the results from the fitted models.
`c.time`	A numeric value providing information on the computational time required to fit all models for each state.
`models`	A data frame listing the models that were fitted.

Examples

```
data(simData)
Y <- simData$Y
init <- Eigen.HMM_init(Y = Y, k = 2, density = "MVT", mod.row = "EEE", mod.col = "EE", nstartR = 10)
fit <- Eigen.HMM_fit(Y = Y, init.par = init, nThreads = 1)
```

`Eigen.HMM_init` *Initialization for ECM Algorithms in Matrix-Variate Hidden Markov Models*

Description

Initializes the ECM algorithms used for fitting parsimonious matrix-variate Hidden Markov Models (HMMs). Parallel computing is implemented and highly recommended for faster computations.

Usage

```
Eigen.HMM_init(
  Y,
  k,
  density,
  mod.row = "all",
  mod.col = "all",
  nstartR = 50,
  nThreads = 1,
  verbose = FALSE,
  seed = 3
)
```

Arguments

`Y`	An array with dimensions `p` x `r` x `num` x `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, `num` is the number of data observations, and `t` is the number of time points.
`k`	An integer or vector indicating the number of states in the model(s).
`density`	A character string specifying the distribution to use in the HMM. Possible values are: "MVN" for the matrix-variate normal distribution, "MVT" for the matrix-variate t-distribution, and "MVCN" for the matrix-variate contaminated normal distribution.
`mod.row`	A character string indicating the parsimonious structure of the row covariance (or scale) matrices. Possible values are: "EII", "VII", "EEI", "VEI", "EVI", "VVI", "EEE", "VEE", "EVE", "EEV", "VVE", "VEV", "EVV", "VVV", or "all". When "all" is specified, all 14 parsimonious structures are considered.

`mod.col`	A character string indicating the parsimonious structure of the column covariance (or scale) matrices. Possible values are: "II", "EI", "VI", "EE", "VE", "EV", "VV", or "all". When "all" is specified, all 7 parsimonious structures are considered.
`nstartR`	An integer specifying the number of random starts to consider.
`nThreads`	A positive integer indicating the number of cores to use for parallel processing.
`verbose`	A logical value indicating whether to display the running output.
`seed`	A positive integer specifying the seed for random generation.

Value

A list containing the following elements:

`results`	A list of the results from the initialization.
`k`	The number of states fitted in each model.
`req.model`	A data frame listing the models that were initialized.
`init.used`	A data frame listing the initializations used for the required models.
`index`	A numeric vector to be used by the `Eigen.HMM_fit()` function.
`dens`	The density used for the HMMs.

Examples

```
data(simData)
Y <- simData$Y
init <- Eigen.HMM_init(Y = Y, k = 2, density = "MVT", mod.row = "EEE", mod.col = "EE", nstartR = 10)
```

`extract.bestM` *Selection of the best fitting model(s)*

Description

This functions extracts the best fitting model(s) according to the Bayesian information criterion (BIC).

Usage

```
extract.bestM(results, top = 1)
```

Arguments

`results`	The output of the `Eigen.HMM_fit()` function.
`top`	Integer. Specifies the number of top-ranked models to display based on the Bayesian Information Criterion (BIC).

Value

A list containing the required best fitting model(s).

Examples

```
data(simData)
Y <- simData$Y
init <- Eigen.HMM_init(Y = Y, k = 2, density = "MVT", mod.row = "EEE", mod.col = "EE", nstartR = 10)
fit <- Eigen.HMM_fit(Y = Y, init.par = init, nThreads = 1)
win <- extract.bestM(results = fit, top = 1)
```

`r.HMM` *Random Number Generation for Matrix-Variate Hidden Markov Models*

Description

Generates random numbers for matrix-variate Hidden Markov Models (HMMs) based on matrix-variate normal, t, and contaminated normal distributions.

Usage

```
r.HMM(density, num, t, PI, M, U, V, IP, nu, alpha, eta)
```

Arguments

`density`	A character string specifying the distribution to use for the HMM. Possible values are: "MVN" for the matrix-variate normal distribution, "MVT" for the matrix-variate t-distribution, and "MVCN" for the matrix-variate contaminated normal distribution.
`num`	An integer specifying the number of random matrices to generate.
`t`	An integer specifying the number of time points.
`PI`	A matrix representing the transition probability matrix.
`M`	An array with dimensions `p` x `r` x `k`, where `k` is the number of states, containing the mean matrices.
`U`	An array with dimensions `p` x `p` x `k`, where `k` is the number of states, containing the row covariance (scale) matrices.
`V`	An array with dimensions `r` x `r` x `k`, where `k` is the number of states, containing the column covariance (scale) matrices.
`IP`	A numeric vector of length `k` containing the initial probability weights.
`nu`	A numeric vector of length `k` containing the degrees of freedom for each state in the MVT distribution.
`alpha`	A numeric vector of length `k` containing the proportion of typical points in each state for the MVCN distribution.
`eta`	A numeric vector of length `k` containing the inflation parameters for each state in the MVCN distribution.

Value

A list containing the following elements:

`Y`	An array with dimensions `p` x `r` x `num` x `t` containing the generated data.
`obs.states`	An `num` x `t` matrix containing the state memberships.

Examples

```
p <- 2
r <- 3
num <- 50
t <- 3
k <- 2
IP <- c(0.5, 0.5)
PI <- matrix(c(0.9, 0.1, 0.3, 0.7), nrow = k, ncol = k, byrow = TRUE)
M <- array(NA, dim = c(p, r, k))
M[,,1]<- matrix(c(0,1,1,
                 -1,-1.5,-1),nrow = p, ncol = r, byrow = TRUE)
M[,,2]<- M[,,1]+3
U <- array(NA, dim = c(p, p, k))
V <- array(NA, dim = c(r, r, k))
U[, , 1] <- U[, , 2] <- matrix(c(1.73, -0.59, -0.59, 2.52), nrow = p, ncol = p, byrow = TRUE)
V[, , 1] <- V[, , 2] <- matrix(c(0.69, 0.23, -0.03,
                                 0.23, 0.48,  0.16,
                                -0.03, 0.16,  0.88), nrow = r, ncol = r, byrow = TRUE)
nu <- c(4.5, 6.5)
simData <- r.HMM(density = "MVT", num = num, t = t, PI = PI,
                M = M, U = U, V = V, IP = IP, nu = nu)
```

`simData` *A Simulated Dataset from a Matrix-Variate t Hidden Markov Model*

Description

A simulated dataset generated from a matrix-variate t Hidden Markov Model with 2 states and an EE - EE covariance structure.

Usage

```
data(simData)
```

Format

A list containing two elements:

1) An array with `p = 2` variables in the rows, `r = 3` variables in the columns, `num = 50` matrices, and `t = 3` time points.

2) An `num` x `t` matrix containing the state memberships.

`simData2` *A Simulated Dataset with Atypical Matrices*

Description

A simulated dataset containing atypical matrices. The data are initially generated from a matrix-variate normal Hidden Markov Model with 2 states and an EE - EE covariance structure. Atypical matrices are then introduced by randomly replacing some of the original matrices with values from a uniform distribution.

Usage

```
data(simData2)
```

Format

A list containing three elements:

1) An array with `p = 2` variables in the rows, `r = 3` variables in the columns, `num = 50` matrices, and `t = 3` time points.

2) An `num` x `t` matrix containing the state memberships.

3) An `num` x `t` matrix identifying the atypical matrices, where atypical matrices are coded with a 1.

Index