

Package ‘MHMM’

March 20, 2020

Type Package

Title Finite Mixture of Hidden Markov Model

Version 1.0.0

Date 2020-03-20

Author Marie Du-Roy-De-Chaumary, Matthieu Marbac and Fabien Navarro

Maintainer Matthieu Marbac <matthieu.marbac-lourdelle@ensai.fr>

Description

Estimation of the latent states and partition by maximum likelihood. Model can be used for analyzing accelerometer data. In such a case, the latent states corresponds to activity levels and the partition permits to consider heterogeneity within the population. Emission laws are zero-inflated gamma distributions. Their parameters depends on the latent states but not on the partition, to compare the time spent by activity levels between classes. Model description is available in Du Roy de Chaumaray, M. and Marbac, M. and Navarro, F. (2019) <arXiv:1906.01547>.

License GPL (>= 2)

Depends R (>= 3.4.4)

Imports Rcpp (>= 0.11.1), methods, parallel, ggplot2, reshape2, gridExtra

LinkingTo Rcpp, RcppArmadillo, BH

Collate 'param.R' 'checkinputs.R' 'rmodel.R' 'data.R' 'MHMM.R'
'results.R' 'plot.R' 'smartinit.R' 'viterbi.R' 'MHMMpackage.R'
'RcppExports.R'

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-03-20 14:40:02 UTC

R topics documented:

MHMM-package	2
accelero	3
mhmm	3
mhmmdata-class	4
mhmmparam-class	5
mhmmresults-class	5
plot	6
rdata.mhmm	7

Index	8
--------------	----------

MHMM-package	<i>Finite Mixture of Hidden Markov Models for accelerometer data</i>
--------------	--

Description

Package:	MHMM
Type:	Package
Version:	1.0.0
Date:	2020-03-20
License:	GPL-2
LazyLoad:	yes

References

Du Roy de Chaumaray, M. and Marbac, M. and Navarro, F. (2019). Mixture of hidden Markov models for accelerometer data. arXiv preprint arXiv:1906.01547

Examples

```
data(accelero)
# To make the estimation <5
res <- mhmm(accelero, K = 2, M = 4, nbcores = 1, nbinit = 5, iterSmall = 2)
plot(res, 1)
```

```
data(accelero)
# It is better to increase the number of random initializations
res <- mhmm(accelero, K = 2, M = 4, nbcores = 1)
plot(res, 1)
```

accelero	<i>Accelerometer data</i>
----------	---------------------------

Description

Accelerometer data measured each 5 minutes on three subjects

References

Huang, Q., Cohen, D., Komarzynski, S., Li, X.-M., Innominato, P., Lévi, F., and Finkenstädt, B. (2018b). Hidden markov models for monitoring circadian rhythmicity in telemetric activity data. *Journal of The Royal Society Interface*, 15(139):20170885

Examples

```
data(accelero)
```

mhmm	<i>Mixture model of Hidden Markov Models.</i>
------	---

Description

This function performs maximum likelihood inference of mixture of hidden Markov models

Usage

```
mhmm(y, K, M, smartinit = TRUE, nbinit = 100, tol = 10(-4),
      nbKeep = min(20, nbinit), iterSmall = 10, nbcores = 1)
```

Arguments

y	list. Observations (one element of the list per subject, missing values should be encoded by NA).
K	numeric. Number of mixture components.
M	numeric. Number of states.
smartinit	boolean. If TRUE, MLE of the mixture neglecting the time dependency is used for the initialization of the algorithm.
nbinit	numeric. Number of initializations.
tol	numeric. It indicates the maximal gap between two successive iterations of EM algorithm which stops the algorithm
nbKeep	numeric. It indicates the number of chains used for the final EM algorithm
iterSmall	numeric. It indicates the number of iterations for each SmallEM algorithm
nbcores	numeric. It defines the number of cores used by the algorithm

Value

Returns an instance of [mhmmresults](#).

References

Du Roy de Chaumaray, M. and Marbac, M. and Navarro, F. (2019). Mixture of hidden Markov models for accelerometer data. arXiv preprint arXiv:1906.01547

Examples

```
data(accelero)
# To make the estimation <5
res <- mhmm(accelero, K = 2, M = 4, nbcores = 1, nbinit = 5, iterSmall = 2)
plot(res, 1)
```

```
data(accelero)
# It is better to increase the number of random initializations
res <- mhmm(accelero, K = 2, M = 4, nbcores = 1)
plot(res, 1)
```

mhmmdata-class

Constructor of [mhmmdata](#) class

Description

nobs numeric. number of subjects

yi list. each element corresponds to the sequences of a single subject.

nbseq nbseq. number of sequences for each subject.

nbtimeobs list. length of each sequence.

tstart list. starting time of each sequence.

Examples

```
getSlots("mhmmdata")
```

mhmmparam-class *Constructor of mhmmparam class*

Description

- K** numeric. Number of classes.
- M** numeric. Number of latent states (activity levels).
- A** list. Matrices of the transition probabilities for each class.
- delta** numeric. Proportions of the classes.
- pi** matrix. Probabilities of the latent states per class (stationary distribution of the Markov chains).
- lambda** list. Parameters of the emission laws (eps: proportions of the zero-inflated component, a: shapes of the gamma distributions, b: rates of the gamma distributions)

Examples

```
getSlots("mhmmparam")
```

mhmmresults-class *Constructor of mhmmresults class*

Description

- param** mhmmparam. MLE of the model parameters
- data** mhmmdata. Data
- partitions** list. Elements of the latent variables (partition among subjects and latent states)
- probabilities** list. Posterior probabilities of the latent variables.
- meantimesperstates** matrix. Mean time spent by each subject (row) into each activity level (column).
- meanvalueperstates** numeric. Summary statistics of the states.
- loglike** numeric. Loglikelihood.
- bic** numeric. BIC.

Examples

```
getSlots("mhmmresults")
```

plot

Plots of an instance of `mhmmresults`

Description

Plots of an instance of `mhmmresults`

Usage

```
## S4 method for signature 'mhmmresults,numeric'  
plot(x, y,  
     col = x@partitions$states[[y]], xlab = "Time", ylab = "Activity",  
     ylim = range(na.omit(x@data@yi[[y]])))
```

Arguments

<code>x</code>	instance of <code>mhmmresults</code> .
<code>y</code>	numeric index of the subject to visualize.
<code>col</code>	numeric indicates the latent state at each time (length must be equal to the length of <code>x</code>)
<code>xlab</code>	character label of the x-axis
<code>ylab</code>	character label of the y-axis
<code>ylim</code>	numeric range of the y-axis

Examples

```
data(accelero)  
# To make the estimation <5  
res <- mhmm(accelero, K = 2, M = 4, nbcores = 1, nbinit = 5, iterSmall = 2)  
plot(res, 1)
```

```
data(accelero)  
# It is better to increase the number of random initializations  
res <- mhmm(accelero, K = 2, M = 4, nbcores = 1)  
plot(res, 1)
```

`rdata.mhmm`*Generator*

Description

This function generates sequence from a MHMM model

Usage

```
rdata.mhmm(n = 50, nbT = 40, A = list(matrix(c(0.7, 0.3, 0.1, 0.2,
0.6, 0.1, 0.1, 0.1, 0.8), 3, 3), matrix(1/3, 3, 3)),
delta = rep(1/length(A), length(A)), a = c(1, 2, 3) * 10, b = c(1,
1, 1), eps = c(0.5, 0.1, 0.2) * 0)
```

Arguments

<code>n</code>	numeric, number of subjects.
<code>nbT</code>	numeric, length of the sequence.
<code>A</code>	list, matrices of the transition probabilities per class
<code>delta</code>	numeric, proportions of the classes.
<code>a</code>	numeric, shapes of the gamma distributions.
<code>b</code>	numeric, rates of the gamma distributions.
<code>eps</code>	numeric, proportions of the zero-inflated in the emission laws.

Value

Returns a list of sequences.

Examples

```
ech <- rdata.mhmm(25, 10)
res <- mhmm(ech$y, K = 2, M = 4, nbcores = 1, nbinit = 5, iterSmall = 2)
```

Index

*Topic **datasets**

accelero, [3](#)

*Topic **package**

MHMM-package, [2](#)

accelero, [3](#)

MHMM (MHMM-package), [2](#)

mhmm, [3](#)

MHMM-package, [2](#)

mhmmdata, [4](#), [5](#)

mhmmdata-class, [4](#)

mhmmparam, [5](#)

mhmmparam-class, [5](#)

mhmmresults, [4-6](#)

mhmmresults-class, [5](#)

plot, [6](#)

plot,mhmmresults,ANY-method (plot), [6](#)

plot,mhmmresults,numeric-method (plot),

[6](#)

plot,mhmmresults-method (plot), [6](#)

rdata.mhmm, [7](#)