

Package ‘funLBM’

July 27, 2018

Type Package

Title Model-Based Co-Clustering of Functional Data

Version 1.0

Date 2018-07-18

Author Charles Bouveyron and Julien Jacques

Maintainer Charles Bouveyron <charles.bouveyron@math.cnrs.fr>

Depends fda, parallel, funFEM, R (>= 3.1.0)

Description

The funLBM algorithm allows to simultaneously cluster the rows and the columns of a data matrix where each entry of the matrix is a function or a time series.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-27 21:10:11 UTC

R topics documented:

ari	2
funLBM	2
plot.funLBM	4
print.funLBM	5
simulateData	6
Velib	7
Index	8

ari	<i>Adjusted Rand index</i>
-----	----------------------------

Description

The adjusted Rand index (ARI) allows to compare two clustering partitions.

Usage

```
ari(x, y)
```

Arguments

x	The first partition to compare (as vector).
y	The second partition to compare (as vector).

Value

ari	The value of the ARI.
-----	-----------------------

See Also

[funLBM](#)

Examples

```
x = sample(1:3,20,replace = TRUE)
y = sample(1:3,20,replace = TRUE)
ari(x,y)
```

funLBM	<i>The functional latent block model</i>
--------	--

Description

The funLBM algorithm allows to simultaneously cluster the rows and the columns of a data matrix where each entry of the matrix is a function or a time series.

Usage

```
funLBM(X, K, L, maxit = 50, burn = 25, basis.name = "fourier", nbasis = 15,
        nbinit = 1, gibbs.it = 3, display = FALSE, init = "kmeans", mc.cores = 1, ...)
```

Arguments

X	The data array ($n \times p \times T$) where each entry corresponds to the measure of one individual i , $i=1,\dots,n$, for a functional variable j , $j=1,\dots,p$, at point t , $t=1,\dots,T$,
K	The number of row clusters,
L	The number of column clusters,
maxit	The maximum number of iterations of the SEM-Gibbs algorithm (default is 100),
burn	The number of iterations of the burn-in period (default is 50),
basis.name	The name ('fourier' or 'spline') of the basis functions used for the decomposition of the functions (default is 'fourier'),
nbasis	Number of the basis functions used for the decomposition of the functions (default is 15),
nbinit	Number of initializations (default is 3),
gibbs.it	Number of Gibbs iterations (default is 3),
display	Binary value. If TRUE, information about the iterations is displayed,
init	The type of initialization: 'random', 'kmeans' or 'funFEM'. Default is 'kmeans',
mc.cores	The number of cores for parallel computing (default is 1),
...	Additional parameters to provide to sub-functions.

Value

The resulting object contains, in addition to call information:

prms	A list containing all fitted parameters for the best model (according to ICL)
Z	The dummy matrix of row clustering
W	The dummy matrix of column clustering
row_clust	The group memberships of rows
col_clust	The group memberships of columns
allPrms	A list containing the fitted parameters for all tested models
loglik	The log-likelihood of the best model
icl	The value of ICL for the best model

References

C. Bouveyron, L. Bozzi, J. Jacques and F.-X. Jollois, The Functional Latent Block Model for the Co-Clustering of Electricity Consumption Curves, *Journal of the Royal Statistical Society, Series C*, 2018 (<https://doi.org/10.1111/rssc.12260>).

Examples

```
## Co-clustering on simulated data with K=4 and L=3
set.seed(12345)
X = simulateData(n = 30, p = 30, t = 15)
out = funLBM(X$data,K=4,L=3)

# Visualization of results
plot(out,type='blocks')
plot(out,type='proportions')
plot(out,type='means')

# Evaluating clustering results
ari(out$col_clust,X$col_clust)
ari(out$row_clust,X$row_clust)

## Co-clustering on simulated data with parallel model selection
## To enable parallel computation (on Linux and Mac systems),
## set mc.cores to the number of available CPUs
## and increase nbinit to allow the best possible fit
set.seed(12345)
X = simulateData(n = 30, p = 30, t = 15)
out = funLBM(X$data,K=2:4,L=2:4,init = "funFEM",mc.cores = 1,nbinit = 1)

# Evaluating clustering results
ari(out$col_clust,X$col_clust)
ari(out$row_clust,X$row_clust)

## Co-clustering of Velib data
data(Velib)
out = funLBM(Velib$data,K=4,L=2)

# Visualization of results
plot(out,type='blocks')
plot(out,type='proportions')
plot(out,type='means')
```

plot.funLBM

Plotting co-clustering results of funLBM

Description

Plotting of funLBM co-clustering results: functional means, block matrix, parameters, ...

Usage

```
## S3 method for class 'funLBM'
plot(x,type='blocks',...)
```

Arguments

x An object produced by the funLBM function,
 type The type of plot to display. Possible plots are 'blocks' (default), 'means', 'evolution', 'criterion', 'likelihood', 'proportions',
 ... Additional arguments to provide.

See Also

[funLBM](#)

Examples

```
## Co-clustering of the Velib data
X = simulateData(n = 30, p = 30, t = 15)
out = funLBM(X$data,K=4,L=3)

# Visualization of results
plot(out,type='blocks')
plot(out,type='proportions')
plot(out,type='means')
```

print.funLBM

Printing co-clustering results of funLBM

Description

Printing a summary of the funLBM co-clustering results

Usage

```
## S3 method for class 'funLBM'
print(x,...)
```

Arguments

x An object produced by the funLBM function,
 ... Additional arguments to provide.

See Also

[funLBM](#)

Examples

```
## Co-clustering of the Velib data
X = simulateData(n = 30, p = 30, t = 15)
out = funLBM(X$data,K=4,L=3)
out
```

simulateData	<i>Simulate data for funLBM</i>
--------------	---------------------------------

Description

Simulate data according to the funLBM model with $K=4$ groups for rows and $L=3$ groups for columns.

Usage

```
simulateData(n = 100, p = 100, t = 30)
```

Arguments

n	The number of rows (individuals) of the simulated data array,
p	The number of columns (functional variables) of the simulated data array,
t	The number of measures for the functions of the simulated data array.

Value

The resulting object contains:

data	data array of size $n \times p \times t$
row_clust	Group memberships of rows
col_clust	Group memberships of columns

References

C. Bouveyron, L. Bozzi, J. Jacques and F.-X. Jollois, The Functional Latent Block Model for the Co-Clustering of Electricity Consumption Curves, Journal of the Royal Statistical Society, Series C, 2018 (<https://doi.org/10.1111/rssc.12260>).

See Also

[funLBM](#)

Examples

```
# Simulate data and co-clustering
set.seed(12345)
X = simulateData(n = 30, p = 30, t = 15)
out = funLBM(X$data, K=4, L=3)

# Visualization of results
plot(out, type='blocks')
plot(out, type='proportions')
plot(out, type='means')
```

```
# Evaluating clustering results
ari(out$col_clust,X$col_clust)
ari(out$row_clust,X$row_clust)
```

Velib

The Velib data set.

Description

The Velib data set contains data from the bike sharing system of Paris, called Velib. The data are loading profiles of the bike stations over seven days. The data were collected every hour during the period Sunday 1st Sept. - Sunday 7th Sept., 2014.

Usage

```
data("Velib")
```

Format

The format is: - data: the loading profiles (nb of available bikes / nb of bike docks) of the 1189 stations for 7 days every hour. - position: the longitude and latitude of the 1189 bike stations.

Source

The real time data are available at <https://developer.jcdecaux.com/> (with an api key).

References

The data were first used in C. Bouveyron, E. Come and J. Jacques, The discriminative functional mixture model for a comparative analysis of bike sharing systems, *The Annals of Applied Statistics*, vol. 9 (4), pp. 1726-1760, 2015 (<http://dx.doi.org/10.1214/15-AOAS861>).

Examples

```
data(Velib)

# Co-clustering
out = funLBM(Velib$data,K=4,L=2)

# Visualization of results
plot(out,type='blocks')
plot(out,type='proportions')
plot(out,type='means')
```

Index

*Topic **Clustering**

funLBM, [2](#)

*Topic **Functional data**

funLBM, [2](#)

*Topic **datasets**

Velib, [7](#)

ari, [2](#)

funLBM, [2](#), [2](#), [5](#), [6](#)

plot.funLBM, [4](#)

print.funLBM, [5](#)

simulateData, [6](#)

Velib, [7](#)