Package 'MatTransMix'

September 11, 2020

Version 0.1.12

Date 2020-09-09

Title Clustering with Matrix Gaussian and Matrix Transformation Mixture Models

Depends R (>= 3.0.0)

LazyLoad yes

LazyData no

Description Provides matrix Gaussian mixture models, matrix transformation mixture models and their model-based clustering results. The parsimonious models of the mean matrices and variance covariance matrices are implemented with a total of 196 variations.

License GPL (>= 2)

Imports mvtnorm

Author Xuwen Zhu [aut, cre],
Volodymyr Melnykov [aut],
Shuchismita Sarkar [ctb],
Michael Hutt [ctb, cph],
Stephen Moshier [ctb, cph],
Rouben Rostamian [ctb, cph],
Carl Edward Rasmussen [ctb, cph],
Dianne Cook [ctb, cph]

Maintainer Xuwen Zhu <xzhu20@cba.ua.edu>

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-09-11 16:20:02 UTC

R topics documented:

`MatTransMix-package` *Finite mixture modeling and model-based clustering of matrices based on matrix Gaussian mixture and matrix transformation mixture models.*

Description

The utility of this package is the clustering of random matrices. Finite mixture modeling and model-based clustering based on matrix Gaussian mixtures and matrix transformation mixtures are employed.

Details

Package:	MatTransMix
Type:	Package
Version:	0.1.1
Date:	2017-02-09
License:	GPL (>= 2)
LazyLoad:	no

Function 'MatTrans.init' runs the initialization for the EM algorithm.

Function 'MatTrans.EM' runs the EM algorithm for matrix-variate mixtures to cluster matrices.

Author(s)

Xuwen Zhu and Volodymyr Melnykov.

Maintainer: Xuwen Zhu <xuwen.zhu@louisville.edu>

`crime` *Crime data*

Description

Data collected by FBI's Uniform Crime on the violent and property crimes of 236 cities.

Usage

```
data(crime)
```

Format

A list of 3 objects: Y, department and state. Y represents the crime rate data array from 236 cities. Department is the police department names and state represents the states where each city is located at. Y is of dimensionality 10 x 13 x 236 with 236 crime rates on the following 10 variables from year 2000 through 2012.

Population Population of each city;

Violent Crime rate Total number of violent crimes;

Murder and non-negligent manslaughter rate Number of murders;

Forcible rape rate Number of rape crimes;

Robbery rate Number of robberies;

Aggravated assault rate Number of assaults;

Property crime rate Total number of property crimes;

Burglary rate Number of burglary crimes;

Larceny-theft rate Number of theft crimes;

Motor vehicle theft rate Number of vehicle theft crimes;

Details

The data have been made publicly available by FBI's Uniform Crime Reports.

Examples

```
data(crime)
```

IMDb *IMDb data*

Description

Data collected from IMDb.com on the ratings of 105 popular comedy movies.

Usage

```
data(IMDb)
```

Format

A list of 2 objects: Y and name, where Y represents the data array of ratings and name represents the comedy movie names. Y is the of dimensionality 2 x 4 x 105 with ratings on 105 movies from female and male by age groups 0-18, 18-29, 30-44, 45+.

Details

The data are publicly available on www.IMDb.com.

Examples

```
data(IMDb)
```

`MatTrans.EM` *EM algorithm for matrix clustering*

Description

Runs the EM algorithm for matrix clustering

Usage

```
MatTrans.EM(Y, initial = NULL, la = NULL, nu = NULL,
model = NULL, trans = "None", la.type = 0,
row.skew = TRUE, col.skew = TRUE, tol = 1e-05,
short.iter = NULL, long.iter = 1000, all.models = TRUE,
size.control = 0, silent = TRUE)
```

Arguments

`Y`	dataset of random matrices (p x T x n), n random matrices of dimensionality (p x T)
`initial`	initialization parameters provided by function MatTrans.init()
`la`	initial skewness for rows (K x p)
`nu`	initial skewness for columns (K x T)
`model`	parsimonious model type, if null, then all 196 models are run
`trans`	transformation method: None (Gaussian models), Power, Manly
`la.type`	lambda type 0 or 1, 0: unrestricted, 1: same lambda across all variables
`row.skew`	if skewness for rows are fitted: TRUE or FALSE
`col.skew`	if skewness for columns are fitted: TRUE or FALSE
`tol`	tolerance level
`short.iter`	number of short EM iterations; if not specified, just run long EM
`long.iter`	number of long EM iterations
`all.models`	if true, run long EM for all models; otherwise just the best model returned by short EM in terms of BIC
`size.control`	minimum size of clusters allowed for controlling spurious solutions
`silent`	whether to produce output of steps or not

Details

Runs the EM algorithm for modeling and clustering matrices for a provided dataset. Both matrix Gaussian mixture, matrix Power mixture and matrix Manly transformation mixture can be employed. The user should use the MatTrans.init() function to get initial parameters and input them as 'initial'. In the case when transformation parameters are not provided but 'trans' is specified to be 'Power' or 'Manly', 'la' and 'nu' take value of 0.5. 'model' can be specified as 'X-XXX-XX'. The first digit 'X' stands for the mean structure. It is either 'G': general mean or 'A': additive mean. The second 'XXX' specifies the variance-covariance Sigma. There are 14 options including EII, VII, EEI, VEI, EVI, VVI, EEE, EVE, VEE, VVE, EEV, VEV, EVV and VVV with detailed explanation as follows: "EII" spherical, equal volume "VII" spherical, unequal volume "EEI" diagonal, equal volume and shape "VEI" diagonal, varying volume, equal shape "EVI" diagonal, equal volume, varying shape "VVI" diagonal, varying volume and shape "EEE" ellipsoidal, equal volume, shape, and orientation "EVE" ellipsoidal, equal volume and orientation (*) "VEE" ellipsoidal, equal shape and orientation (*) "VVE" ellipsoidal, equal orientation (*) "EEV" ellipsoidal, equal volume and equal shape "VEV" ellipsoidal, equal shape "EVV" ellipsoidal, equal volume (*) "VVV" ellipsoidal, varying volume, shape, and orientation The last 2-digit 'XX' specifies the variance-covariance Psi. There are 7 options including II, EI, VI, EE, VE, EV, VV. The user can specify the 'model' to be for example 'X-VVV-EV', then both 'G' and 'A' mean structures will be fitted while Sigma and Psi are fixed at 'VVV' and 'EV', respectively. Similarly, 'model' can be specified as 'G-XXX-EV' or 'G-VVV-XX' for selection of Sigma and Psi structures.

Value

`result`	parsimonious models
`model`	model types
`loglik`	log likelihood values
`bic`	bic values
`best.result`	best parsimonious model
`best.model`	best model type
`best.loglik`	best logliklihood
`best.bic`	best bic

Examples

```
set.seed(123)
data(IMDb)
Y <- IMDb$Y/100
p <- dim(Y)[1]
T <- dim(Y)[2]
n <- dim(Y)[3]
K <- 3
init <- MatTrans.init(Y, K = K, n.start = 2)
M <- MatTrans.EM(Y, initial = init, model = "X-VVV-VV",
long.iter = 1000, silent = FALSE)
```

`MatTrans.init` *Initialization for the EM algorithm for matrix clustering*

Description

Runs the initialization for the EM algorithm for matrix clustering

Usage

```
MatTrans.init(Y, K, n.start = 10, scale = 1)
```

Arguments

`Y`	dataset of random matrices (p x T x n), n random matrices of dimensionality (p x T)
`K`	number of clusters
`n.start`	initial random starts
`scale`	scaling parameter

Details

Random starts are used to obtain different starting values. The number of clusters, the skewness parameters, and number of random starts need to be specified. In the case when transformation parameters are not provided, the function runs the EM algorithm without any transformations, i.e., it is equivalent to the EM algorithm for a matrix Gaussian mixture. Notation: n - sample size, p x T - dimensionality of the random matrices, K - number of mixture components.

Examples

```
set.seed(123)
data(crime)
Y <- crime$Y[c(2,7),,] / 1000
p <- dim(Y)[1]
T <- dim(Y)[2]
n <- dim(Y)[3]
K <- 2
init <- MatTrans.init(Y, K = K, n.start = 2)
```

print.object *Functions for Printing or Summarizing Objects*

Description

EM classes for printing and summarizing objects.

Usage

```
## S3 method for class 'EM'
print(x, ...)
## S3 method for class 'EM'
summary(object, ...)
```

Arguments

x	an object with the 'EM' class attributes.
object	an object with the 'EM' class attributes.
...	other possible options.

Details

Some useful functions for printing and summarizing results.

See Also

MatTrans.EM.

Examples

```
set.seed(123)
data(IMDb)
Y <- IMDb$Y/100
p <- dim(Y)[1]
T <- dim(Y)[2]
n <- dim(Y)[3]
K <- 3
init <- MatTrans.init(Y, K = K, n.start = 2)
M <- MatTrans.EM(Y, initial = init, model = "X-VVV-VV",
long.iter = 1000, silent = FALSE)
print.EM(M)
```

`Salary` *Salary data*

Description

Data collected from the Chronicle of Higher Education web site reporting the average faculty salaries from 696 universities presented in the form of a 2 by 3 by 13 -dimensional tensor.

Usage

```
data(Salary)
```

Format

A list of 2 objects: Y, uni_info. Y represents the salary data array from 696 universities. uni_info has the university names, state and category. Y is of dimensionality 2 by 3 by 13 categorized by the following factors: gender (Male, Female), professor rank (Assistant, Associate, Full), and academic year (2003-2004,2015-2016).

Details

The data have been made publicly available by the Chronicle of Higher Education web site.

Examples

```
data(Salary)
```

Index