Package ‘SpaTimeClus’

December 21, 2016

Type Package

Title Model-Based Clustering of Spatio-Temporal Data

Version 1.0

Date 2016-12-21

Author Cheam A., Marbac M., and McNicholas P.

Maintainer Matthieu Marbac <matthieu.marbac@gmail.com>

Description Mixture model is used to achieve the clustering goal. Each component is itself a mixture model of polynomial autoregressive regressions whose the logistic weights consider the spatial and temporal information.

Imports methods, Rcpp (>= 0.11.1), parallel

LinkingTo Rcpp, RcppArmadillo

License GPL (>= 2)

Depends R (>= 3.0.2)

RoxygenNote 5.0.1

Repository CRAN

Repository/R-Forge/Project spatimeclus

Repository/R-Forge/Revision 19

Repository/R-Forge/DateTimeStamp 2016-12-21 11:08:44

Date/Publication 2016-12-21 17:02:04

NeedsCompilation yes

R topics documented:

`SpaTimeClus-package` *SpaTimeClus a package for clustering spatio-temporal data*

Description

SpaTimeClus is a tool for clustering Spatio-Temporal data.

Details

Package:	SpaTimeClus
Type:	Package
Version:	1.0.0
Date:	2016-12-21
License:	GPL-2
LazyLoad:	yes

The main function of this package is spatimeclus that performs the clustering of spatio-temporal data.

Author(s)

Author: Cheam A., Marbac M., and McNicholas P.

References

Cheam A., Marbac M., and McNicholas P., Model-Based Clustering for Spatio-Temporal Data Applied for Air Quality.

Examples

```
## Not run:
data(airparif)

# Clustering of the data by considering the spatial dependencies
res.spa <- spatimeclus(airparif$obs,  G=3, K=4, Q=4, map = airparif$map,
 nbinitSmall=50, nbinitKept=5, nbiterSmall=5)
summary(res.spa)
```

```
# Clustering of the data without considering the spatial dependencies
res.nospa <- spatimeclus(airparif$obs,  G=3, K=4, Q=4, nbinitSmall=50, nbinitKept=5, nbiterSmall=5)
summary(res.nospa)

## End(Not run)
```

airparif *Real spatio-temporal data: airparif*

Description

airparif$obs describes 101 days in 2015 by indicating the quantity of NO2 at 9 sites around Paris during 24 hours.

Details

airparif$map indicates the locations of the 9 sites around Paris where the measures are taken.

airparif$datasup describes the 101 days with meteorological variables.

Examples

```
data(airparif)
```

BuildSTCdata *Constructors of the class STCdata*

Description

Constructors of the class STCdata

Usage

```
BuildSTCdata(x, map, m = 1:(dim(x)[3]))
```

Arguments

x	array It contains the observations to cluster where the dimesions are respectively: number of the observation, site of the observation, time of the observation.
map	matrix. It gives the spatial coordiantes of each site.
m	numeric. It indicates the moments of observations.

Value

Returns an instance of STCdata.

print *Summary function.*

Description

This function prints the elements of an instance of `STCresults`..

Usage

```
## S4 method for signature 'STCresults'
print(x)
```

Arguments

`x` an instance of `STCresults`.

spatimeclus *This function performs the maximum likelihood estimation for a known model in clustering*

Description

This function performs the maximum likelihood estimation for a known model in clustering

Usage

```
spatimeclus(obs, G, K, Q, map = NULL, m = 1:(dim(obs)[3]), crit = "BIC",
  tol = 0.001, param = NULL, nbcores = 1, nbinitSmall = 500,
  nbinitKept = 50, nbiterSmall = 20, nbiterKept = 500)
```

Arguments

`obs` array It contains the observations to cluster where the dimensions are respectively: number of the observation, site of the observation, time of the observation.

`G` numeric. It defines possible numbers of components.

`K` numeric. It defines possible numbers of regressions per components

`Q` numeric. It defines possible degrees of regressions.

`map` matrix. It gives the spatial coordiantes of each site.

`m` numeric. It indicates the moments of observations (optional, default is 1:T).

`crit` character. It indicates the criterion used for the model selection ("AIC", "BIC" or "ICL", optional, default is "BIC").

`tol` numeric. The algorithm is stopped when the loglikelihood increases less than tol during two successive iterations (optional, default is 0.001).

`param` list of STCparam. It gives the initial values of the EM algorithm (optional, starting point are sampled at random).

`nbcores` numeric. It defines the numerber of cores used by the alogrithm, only for Linux and Mac (optional, default is 1).

`nbinitSmall` numeric. It defines the number of random initializations (optional, default is 500).

`nbinitKept` numeric. It defines the number of chains estimated until convergence (optional, default is 50).

`nbiterSmall` numeric. It defines the number of iterations before keeping the nbinitKept best chains (optional, default is 20).

`nbiterKept` numeric. It defines the maximum number of iterations before to stop the algorith; (optional, default is 500).

Value

Returns an instance of STCresults.

Examples

```
## Not run:
data(airparif)

# Clustering of the data by considering the spatial dependencies
res.spa <- spatimeclus(airparif$obs,  G=3, K=4, Q=4, map = airparif$map,
 nbinitSmall=50, nbinitKept=5, nbiterSmall=5)
summary(res.spa)

# Clustering of the data without considering the spatial dependencies
res.nospa <- spatimeclus(airparif$obs,  G=3, K=4, Q=4, nbinitSmall=50, nbinitKept=5, nbiterSmall=5)
summary(res.nospa)

## End(Not run)
```

`STCcriteria-class` *Constructor of [`STCcriteria`] class*

Description

loglike numeric. It indicates the value of the log-likelihood.

AIC numeric. It indicates the value of the AIC criterion.

BIC numeric. It indicates the value of the BIC criterion.

ICL numeric. It indicates the value of the ICL criterion.

Examples

```
  getSlots("STCcriteria")
```

`STCdata-class` *Constructor of [`STCdata`] class*

Description

x matrix. It contains the observations. Each column corresponds to an individual. The row indicates the values of each site for each time.

TT numeric. It indicates the number of elements of the time grid.

JJ numeric. It indicates the number of sites.

n numeric. It indicates the number of observations.

map numeric. It indicates the spatial coordinates of each site.

Examples

```
getSlots("STCdata")
```

`STCmodel` *This function builts an instance of STCmodel.*

Description

This function builts an instance of STCmodel.

Usage

```
STCmodel(G, K, Q, nospatial)
```

Arguments

`G`	integer. It defines the number of mixture components.
`K`	integer. It defines the number of polynoms for each component.
`Q`	integer. It defines the degree of the polynoms.
`nospatial`	binary. It defines if the spatial dependencies are considered (1:no, 0:yes).

Value

Returns an instance of STCmodel.

STCmodel-class *Constructor of [STCmodel] class*

Description

G integer. It defines the number of mixture components.

K integer. It defines the number of polynoms for each component.

Q integer. It defines the degree of the polynoms.

spatial integer. It defines if the spatial dependencies are considered (1:yes, 0:no).

nbparam integer. It indicates the number of parameters involved by the model.

Examples

```
getSlots("STCmodel")
```

STCparam-class *Constructor of [STCparam] class*

Description

proportions numeric. It defines the component proportions.

lambda list. It defines the logistic coefficients per component.

beta list. It defines the polynom coefficients per component.

sigma matrix. It defines the variance associated to each polynom per component.

Examples

```
getSlots("STCparam")
```

STCpartitions-class *Constructor of [STCpartitions] class*

Description

- **hardind** numeric. It indicates the hard partition of the individuals (obtained by the MAP rule qpplied with the MLE).
- **fuzzyind** matrix. It indicates the fuzzy partition (conditional probability of the component membership) of the individuals.
- **hardseg** list. It indicates the segmentation (most probable polynom according to the spatial and temporal grid) per components

Examples

```
getSlots("STCpartitions")
```

STCresults-class *Constructor of [STCresults] class*

Description

- **model** STCmodel. It contains the elements relied to the model.
- **data** STCdata. It contains the elements relied to the data.
- **param** STCparam. It contains the elements relied to the parameters.
- **criteria** STCcriteria. It contains the elements relied to the information criteria.
- **partitions** STCpartitions. It contains the elements relied to the partitions.
- **tune** STCtune. It contains the tunning parameters of the algorithm.
- **allmodels** matrix. list of the estimnated models and their information criterion.

Examples

```
getSlots("STCresults")
```

STCtune-class *Constructor of [STCtune] class*

Description

tol numeric. The algorithm is stopped when two successive iterations increase the log-likelihood less than tol.

nbinitSmall numeric. Number of random initializations for the short run EM algorithm.

nbinitKept numeric. Number of initializations kept for the long run EM algorithm.

nbiterSmall numeric. Maximum number of iteration before stopping the short run EM algorithm.

nbiterKept numeric. Maximum number of iteration before stopping the long run EM algorithm.

Examples

```
getSlots("STCtune")
```

summary *Summary function.*

Description

This function gives the summary of an instance of STCresults.

Usage

```
## S4 method for signature 'STCresults'
summary(object)
```

Arguments

object	instance of STCresults.

Index