Package 'fdaMocca'

October 21, 2021

Encoding UTF-8

Version 0.1-0

Author Natalya Pya Arnqvist[aut, cre],
Per Arnqvist [aut, cre],
Sara Sjöstedt de Luna [aut]

Maintainer Natalya Pya Arnqvist <nat.pya@gmail.com>

Title Model-Based Clustering for Functional Data with Covariates

Date 2021-10-21

Description Routines for model-based functional cluster analysis for functional data with optional covariates. The idea is to cluster functional subjects (often called functional objects) into homogenous groups by using spline smoothers (for functional data) together with scalar covariates. The spline coefficients and the covariates are modelled as a multivariate Gaussian mixture model, where the number of mixtures corresponds to the number of clusters. The parameters of the model are estimated by maximizing the observed mixture likelihood via an EM algorithm (Arnqvist and Sjöstedt de Luna, 2019) <arXiv:1904.10265>. The clustering method is used to analyze annual lake sediment from lake Kassjön (Northern Sweden) which cover more than 6400 years and can be seen as historical records of weather and climate.

Depends R (>= 3.6.0)

Imports stats, graphics, Matrix, parallel, foreach, doParallel,
mvtnorm, fda, grDevices

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2021-10-21 17:40:04 UTC

R topics documented:

`fdaMocca-package` *Model-based clustering for functional data with covariates*

Description

`fdaMocca` provides functions for model-based functional cluster analysis for functional data with optional covariates. The aim is to cluster a set of independent functional subjects (often called functional objects) into homogenous groups by using basis function representation of the functional data and allowing scalar covariates. A functional subject is defined as a curve and covariates. The spline coefficients and the (potential) covariates are modelled as a multivariate Gaussian mixture model, where the number of mixtures corresponds to the number of (predefined) clusters. The model allows for different cluster covariance structures for the basis coefficients and for the covariates. The parameters of the model are estimated by maximizing the observed mixture likelihood using an EM-type algorithm (Arnqvist and Sjöstedt de Luna, 2019).

Details

Package:	fdaMocca
Type:	Package
License:	GPL (>= 2)

Author(s)

Per Arnqvist, Sara Sjöstedt de Luna, Natalya Pya Arnqvist

Maintainer: Natalya Pya Arnqvist<nat.pya@gmail.com>

References

Arnqvist, P., Bigler, C., Renberg, I., Sjöstedt de Luna, S. (2016). Functional clustering of varved lake sediment to reconstruct past seasonal climate. *Environmental and Ecological Statistics*, **23**(4), 513–529.

Abramowicz, K., Arnqvist, P., Secchi, P., Sjöstedt de Luna, S., Vantini, S., Vitelli, V. (2017). Clustering misaligned dependent curves applied to varved lake sediment for climate reconstruction. *Stochastic Environmental Research and Risk Assessment*. Volume **31.1**, 71–85.

Arnqvist, P., and Sjöstedt de Luna, S. (2019). Model based functional clustering of varved lake sediments. *arXiv preprint arXiv:1904.10265*.

`criteria.mocca` *AIC, BIC, entropy for a functional clustering model*

Description

Function to extract the information criteria AIC and BIC, as well as the average Shannon entropy over all functional objects, for a fitted functional clustering `mocca`. The Shannon entropy is computed over the posterior probability distribution of belonging to a specific cluster given the functional object (see Arnqvist and Sjöstedt de Luna, 2019, for further details).

Usage

```
criteria.mocca(x)
```

Arguments

`x`	fitted model objects of class `mocca` as produced by `mocca()`.

Value

A table with the AIC, BIC and Shannon entropy values of the fitted model.

Author(s)

Per Arnqvist

References

Arnqvist, P., and Sjöstedt de Luna, S. (2019). Model based functional clustering of varved lake sediments. *arXiv preprint arXiv:1904.10265*.

See Also

`logLik.mocca`, `mocca`

Examples

```
## see examples in mocca()
```

`estimate.mocca` *Model parameter estimation*

Description

Function to estimate model parameters by maximizing the observed log likelihood via an EM algorithm. The estimation procedure is based on an algorithm proposed by James and Sugar (2003).

The function is not normally called directly, but rather service routines for `mocca`. See the description of the `mocca` function for more detailed information of arguments.

Usage

```
estimate.mocca(data,K=5,q=6,h=2,random=TRUE,B=NULL,svd=TRUE,
       use.covariates=FALSE,stand.cov=TRUE,index.cov=NULL,
       lambda=1.4e-4,EM.maxit=50, EMstep.tol=1e-8,Mstep.maxit=10,
       Mstep.tol=1e-4, EMplot=TRUE,trace=TRUE,n.cores=NULL)
```

Arguments

`data`	a list containing at least five objects (vectors) named as `x`, `time`, `timeindex`, `curve`, `grid`, `covariates` (optional). See `mocca` for the detailed explanation of each object.
`K`	number of clusters (default: `K=3`).
`q`	number of B-splines used to describe the individual curves. Evenly spaced knots are used (default: `q=6`). (currently only B-splines are implemented, however, it is possible to use other basis functions such as, e.g. Fourier basis functions)
`h`	a positive integer, parameter vector dimension in low-dimensionality representation of the curves (spline coefficients). h should be less or equal to the number of clusters K (default: `h=2`).
`random`	`TRUE/FALSE`, if `TRUE` each subject is randomly assigned to one of the `K` clusters initially, otherwise `k-means` is used to initialize cluster belongings (default: TRUE).
`B`	an Nxq matrix of spline coefficients, the spline approximation of the yearly curves based on p number of splines. If `B=NULL` (default), the coefficients are estimated using `fda::create.bspline.basis`.
`svd`	`TRUE/FALSE`, whether SVD decomposition should be used for the matrix of spline coefficients (default: TRUE).
`use.covariates`	`TRUE/FALSE`, whether covariates should be included when modelling (default: FALSE).
`stand.cov`	`TRUE/FALSE`, whether covariates should be standardized when modelling (default: `TRUE`).

index.cov	a vector of indices indicating which covariates should be used when modelling. If NULL (default) all present covariates are included.
lambda	a positive real number, smoothing parameter value to be used when estimating B-spline coefficients.
EM.maxit	a positive integer which gives the maximum number of iterations for a EM algorithm (default: EM.maxit=50).
EMstep.tol	the tolerance to use within iterative procedure of the EM algorithm (default: EMstep.tol=1e-8).
Mstep.maxit	a positive scalar which gives the maximum number of iterations for an inner loop of the parameter estimation in M step (default: Mstep.maxit=20).
Mstep.tol	the tolerance to use within iterative procedure to estimate model parameters (default: Mstep.tol=1e-4).
EMplot	TRUE/FALSE, whether plots of cluster means with some summary information should be produced at each iteration of the EM algorithm (default: FALSE).
trace	TRUE/FALSE, whether to print the current values of σ^2 and σ_x^2 of the covariates at each iteration of M step (default: TRUE).
n.cores	number of cores to be used with parallel computing.

Value

A list is returned with the following items:

loglik	the maximized log likelihood value.
sig2	estimated residual variance for the spline coefficients (for the model without covariates), or a vector of the estimated residual variances for the spline coefficients and for the covariates (for the model with covariates).
conv	indicates why the EM algorithm terminated: 0: indicates successful completion. 1: indicates that the iteration limit EM.maxit has been reached.
iter	number of iterations of the EM algorithm taken to get convergence.
score.hist	a matrix of the succesive values of the scores: residual variances and log likelihood, up until convergence.
parameters	a list containing all the estimated parameters: $\boldsymbol{\lambda}_0$, $\boldsymbol{\Lambda}$, $\boldsymbol{\alpha}_k$, $\boldsymbol{\Gamma}_k$ (or $\boldsymbol{\Delta}_k$ in presence of covariates), π_k (probabilities of cluster belongnings), σ_x^2 (residual variance for the covariates if present), $\mathbf{v}_k$ (mean values of the covariates for each cluster, in presence of covariates), $k = 1, ..., K$, where K is the number of clusters.
vars	a list containing results from the E step of the algorithm: the posterior probabilities for each subject $\pi_{k\|i}$'s, the expected values of the $\boldsymbol{\gamma}_i$'s, $\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^T$, and the covariance matrix of $\boldsymbol{\gamma}_i$ given cluster membership and the observed values of the curve. See Arnqvist and Sjöstedt de Luna (2019) that explains these values.
data	a list containing all the original data plus re-arranged functional data and covariates (if supplied) needed for EM-steps.

`design` a list of spline basis matrices with and without covariates: `FullS.bmat` is the spline basis matrix $\mathbf{S}$ computed on the grid of uniquily specified time points; `FullS` is the spline basis matrix `FullS.bmat` or $\mathbf{U}$ matrix from the svd of `FullS` (if applied); $\mathbf{S}$ is the spline basis matrix computed on `timeindex`, a vector of time indices from T possible from `grid`; the inverse $(\mathbf{S}^T\mathbf{S})^{-1}$; `tag.S` is the matrix $\mathbf{S}$ with covariates; `tag.FullS` is the matrix `FullS` with covariates.

`initials` a list of initial settings: q is the spline basis dimension, N is the number of objects/curves, Q is the number of basis dimension plus the number of covariates (if present), $random$ is whether k-means was used to initialize cluster belonings, h is the vector dimension in low-dimensionality representation of the curves, K is the number of clusters, r is the number of scalar covariates.

Author(s)

Per Arnqvist, Natalya Pya Arnqvist, Sara Sjöstedt de Luna

References

James, G.M., Sugar, C.A. (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98.462, 397–408.

Arnqvist, P., and Sjöstedt de Luna, S. (2019). Model based functional clustering of varved lake sediments. *arXiv preprint arXiv:1904.10265*.

See Also

`fdaMocca-package`, `mocca`

`logLik.mocca` *Log-likelihood for a functional clustering model*

Description

Function to extract the log-likelihood for a fitted functional clustering `mocca` model (fitted by mixture likelihood maximization).

Note: `estimate.mocca` uses `loglik.EMmocca()` for calculating the log likelihood at each iterative step.

Usage

```
  ## S3 method for class 'mocca'
logLik(object,...)
```

Arguments

`object` fitted model objects of class `mocca` as produced by `mocca()`.

`...` unused in this case

Value

The log-likehood value as `logLik` object.

Author(s)

Per Arnqvist

References

Arnqvist, P., and Sjöstedt de Luna, S. (2019). Model based functional clustering of varved lake sediments. *arXiv preprint arXiv:1904.10265*.

See Also

`estimate.mocca`, `mocca`

`mocca` *Model-based clustering for functional data with covariates*

Description

This function fits a functional clustering model to observed independent functional subjects, where a functional subject consists of a function and possibly a set of covariates. Here, each curve is projected onto a finite dimensional basis and clustering is done on the resulting basis coefficients. However, rather than treating basis coefficients as parameters, mixed effect modelling is used for the coefficients. In the model-based functional clustering approach the functional subjects (i.e. the spline/basis coefficients and the potential covariates) are assumed to follow a multivariate Gaussian mixture model, where the number of distributions in the mixture model corresponds to the number of (predefined) clusters, K. Given that a functional subject belongs to a cluster k, the basis coefficients and covariate values are normally distributed with a cluster-specific mean and covariance structure.

An EM-style algorithm based on James and Sugar (2003) is implemented to fit the Gaussian mixture model for a prespecified number of clusters `K`. The model allows for different cluster covariance structure for the spline coefficients and model coefficients for the covariates. See Arnqvist and Sjöstedt de Luna (2019) for details about differences to the clustering model and its implementation.

The routine calls `estimate.mocca` for the model fitting.

Usage

```
mocca(data=stop("No data supplied"), K = 5, q = 6, h = 2,
     use.covariates=FALSE,stand.cov=TRUE,index.cov=NULL,
     random=TRUE, B=NULL,svd=TRUE,lambda=1.4e-4, EM.maxit=50,
     EMstep.tol=1e-6,Mstep.maxit=20,Mstep.tol=1e-4,EMplot=TRUE,
     trace=FALSE,n.cores=NULL)
```

Arguments

data	a list containing at least three objects (vectors) named as x, time, curve, and optional timeindex, grid and covariates: i) suppose we observe N independent subjects, each consisting of a curve and potentially a set of scalar covariates, where the i^{th} curve has been observed at n_i different time points, $i = 1, ..., N$. x is a vector of length $\sum_{i=1}^{N} n_i$ with the first n_1 elements representing the observations of the first curve, followed by n_2 observations of the second curve, etc; ii) time is a $\sum_i n_i$ vector of the concatenated time points for each curve $(t_{ij}, j = 1, ..., n_i, i = 1, ..., N)$, with the first n_1 elements being the time points at which the first curve is observed, etc. Often, the time points within each curve are scaled to $[0, 1]$. iii) timeindex is a $\sum_i n_i$ vector of time indices from T possible from grid. So each observation has a corresponding location (time index) within $[0, 1]$ uniquely specified time points. If not supplied, obtained from time and grid; iv) curve is a $\sum_i n_i$ vector of integers from $1, ..., N$, specifying the subject number for each observation in x; v) grid is a T vector of all unique time points (values within $[0, 1]$ interval) for all N subjects, needed for estimation of the B-spline coefficients in fda::eval.basis(). timeindex and grid together give the timepoint for each subject (curve). If not supplied, obtained from time. vi) if supplied, covariates is an $N \times r$ matrix (or data frame) of scalar covariates (finite-dimensional covariates).
K	number of clusters (default: K=3).
q	number of B-splines for the individual curves. Evenly spaced knots are used (default: q=6).
h	a positive integer, parameter vector dimension in the low-dimensionality representation of the curves (spline coefficients). h should be smaller than the number of clusters K (default: h=2).
use.covariates	TRUE/FALSE, whether covariates should be used when modelling (default: FALSE).
stand.cov	TRUE/FALSE, whether covariates should be standardized when modelling (default: TRUE).
index.cov	a vector of indices indicating which covariates should be used when modelling. If NULL (default) all present covariates are included.
random	TRUE/FALSE, if TRUE the initial cluster belongings is given by uniform distribution, otherwise k-means is used to initialize cluster belongings (default: TRUE).
B	an $N \times q$ matrix of spline coefficients, the spline approximation of the yearly curves based on p number of splines. If B=NULL (default), the coefficients are estimated using fda:: create.bspline.basis.
svd	TRUE/FALSE, whether SVD decomposition should be used for the matrix of spline coefficients (default: TRUE).
lambda	a positive real number, smoothing parameter value to be used when estimating B-spline coefficients.

`EM.maxit`	a positive integer which gives the maximum number of iterations for a EM algorithm (default: `EM.maxit=50`).
`EMstep.tol`	the tolerance to use within iterative procedure of the EM algorithm (default: `EMstep.tol=1e-8`).
`Mstep.maxit`	a positive scalar which gives the maximum number of iterations for an inner loop of the parameter estimation in M step (default: `Mstep.maxit=20`).
`Mstep.tol`	the tolerance to use within iterative procedure to estimate model parameters (default: Mstep.tol=1e-4).
`EMplot`	`TRUE/FALSE`, whether plots of cluster means with some summary information should be produced at each iteration of the EM algorithm (default: `FALSE`).
`trace`	`TRUE/FALSE`, whether to print the current values of σ^2 and σ_x^2 for the covariates at each iteration of `M step` (default: `FALSE`).
`n.cores`	number of cores to be used with parallel computing. If `NULL` (default) `n.cores` is set to the numbers of available cores - 1 (`n.cores= detectCores()-1`).

Details

A model-based clustering with covariates (mocca) for the functional subjects (curves and potentially covariates) is a gaussian mixture model with K components. Let $g_i(t)$ be the true function (curve) of the i^{th} subject, for a set of N independent subjects. Assume that for each subject we have a vector of observed values of the function $g_i(t)$ at times $t_{i1}, ..., t_{in_i}$, obtained with some measurement errors. We are interested in clustering the subjects into K (homogenous) groups. Let y_{ij} be the observed value of the ith curve at time point t_{ij}. Then

$$y_{ij} = g_i(t_{ij}) + \epsilon_{ij}, i = 1, ..., N, j = 1, ..., n_i,$$

where ϵ_{ij} are assumed to be independent and normally distributed measurement errors with mean 0 and variance σ^2. Let $\mathbf{y}_i$, $\mathbf{g}_i$, and $\boldsymbol{\epsilon}_i$ be the n_i-dimensional vectors for subject i, corresponding to the observed values, true values and measurement errors, respectively. Then, in matrix notation, the above could be written as

$$\mathbf{y}_i = \mathbf{g}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, N,$$

where $\boldsymbol{\epsilon}_i \sim N_{n_i}(\mathbf{0}, \sigma^2 \mathbf{I}_{n_i})$. We further assume that the smooth function $g_i(t)$ can be expressed as

$$g_i(t) = \boldsymbol{\phi}^T(t)\boldsymbol{\eta}_i,$$

where $\boldsymbol{\phi}(t) = (\phi_1(t), \ldots, \phi_p(t))^T$ is a p-dimensional vector of known basis functions evaluated at time t, e.g. B-splines, and $\boldsymbol{\eta}_i$ a p-dimensional vector of unknown (random) coefficients. The $\boldsymbol{\eta}_i$'s are modelled as

$$\boldsymbol{\eta}_i = \boldsymbol{\mu}_{z_i} + \boldsymbol{\gamma}_i, \quad \boldsymbol{\eta}_i \sim N_p(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Gamma}_{z_i}),$$

where $\boldsymbol{\mu}_{z_i}$ is a vector of expected spline coefficients for a cluster k and z_i denotes the unknown cluster membership, with $P(z_i = k) = \pi_k$, $k = 1, \ldots, K$. The random vector $\boldsymbol{\gamma}_i$ corresponds to subject-specific within-cluster variability. Note that this variability is allowed to be different in different clusters, due to $\boldsymbol{\Gamma}_{z_i}$. If desirable, given that subject i belongs to cluster $z_i = k$, a further parametrization of $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$, may prove useful, for producing low-dimensional representations of the curves as suggested by James and Sugar (2003):

$$\boldsymbol{\mu}_k = \boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k,$$

where $\boldsymbol{\lambda}_0$ and $\boldsymbol{\alpha}_k$ are p- and h-dimensional vectors respectively, and $\boldsymbol{\Lambda}$ is a $p \times h$ matrix with $h \leq K-1$. Choosing $h < K-1$ may be valuable, especially for sparse data. In order to ensure identifiability, some restrictions need to be put on the parameters. Imposing the restriction that $\sum_{k=1}^{K} \boldsymbol{\alpha}_k = \mathbf{0}$ implies that $\boldsymbol{\phi}^T(t)\boldsymbol{\lambda}_0$ can be viewed as the overall mean curve. Depending on the choice of h, p and K, further restrictions may need to be imposed in order to have identifiability of the parameters ($\boldsymbol{\lambda}_0, \boldsymbol{\Gamma}$ and $\boldsymbol{\alpha}_k$ are confounded if no restrictions are imposed). In vector-notation we thus have

$$\mathbf{y}_i = \mathbf{B}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{z_i} + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i, \;\; i = 1, ..., N,$$

where $\mathbf{B}_i$ is an $n_i \times p$ matrix with $\boldsymbol{\phi}^T(t_{ij})$ on the j^{th} row, $j = 1, \ldots, n_i$. We will also assume that the $\boldsymbol{\gamma}_i$'s, $\boldsymbol{\epsilon}_i$'s and the z_i's are independent. Hence, given that subject i belongs to cluster $z_i = k$ we have

$$\mathbf{y}_i | z_i = k \;\; \sim \;\; N_{n_i}\left(\mathbf{B}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k), \;\; \mathbf{B}_i\boldsymbol{\Gamma}_k\mathbf{B}_i^T + \sigma^2\mathbf{I}_{n_i}\right).$$

Based on the observed data $\mathbf{y}_1, \ldots, \mathbf{y}_N$, the parameters $\boldsymbol{\theta}$ of the model can be estimated by maximizing the observed likelihood

$$L_o(\boldsymbol{\theta}|\mathbf{y}_1, \ldots, \mathbf{y}_N) = \prod_{i=1}^{N}\sum_{k=1}^{G}\pi_k f_k(\mathbf{y}_i, \boldsymbol{\theta}),$$

where

$\boldsymbol{\theta} = \left\{\boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_K, \pi_1, \ldots, \pi_K, \sigma^2, \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_K\right\}$, and $f_k(\mathbf{y}_i, \boldsymbol{\theta})$ is the normal density given above. Note that here $\boldsymbol{\theta}$ will denote all scalar, vectors and matrices of parameters to be estimated. An EM-type algorithm is used to maximize the likelihood above.

If additional covariates have been observed for each subject besides the curves, they can also be included in the model when clustering the subjects. Given that the subject i belongs to cluster $k, (z_i = k)$ the r covariates $\boldsymbol{x}_i \in \mathbf{R}^r$ are assumed to have mean value $\boldsymbol{\upsilon}_k$ and moreover $\boldsymbol{x}_i = \boldsymbol{\upsilon}_k + \boldsymbol{\delta}_i + \boldsymbol{e}_i$, where we assume that $\boldsymbol{\delta}_i | z_i = k \sim N_r(\mathbf{0}, \mathbf{D}_k)$ is the random deviation within cluster and $\boldsymbol{e}_i \sim N_r(\mathbf{0}, \sigma_x^2\mathbf{I}_r)$ independent remaining unexplained variability. Note that this model also incorporates the dependence between covariates and the random curves via the random basis coefficients. See Arnqvist and Sjöstedt de Luna (2019) for further details. EM-algorithm is implemented to maximize the mixture likelihood.

The method is applied to annually varved lake sediment data from the lake Kassjön in Northern Sweden. See an example and also `varve` for the data description.

Value

The function returns an object of class `"mocca"` with the following elements:

`loglik`	the maximized log likelihood value.
`sig2`	estimated residual variance for the functional data (for the model without covariates), or a vector of the estimated residual variances for the functional data and for the covariates (for the model with covariates).
`conv`	indicates why the EM algorithm terminated: 0: indicates successful completion. 1: indicates that the iteration limit `EM.maxit` has been reached.
`iter`	number of iterations of the EM algorithm taken to get convergence.

`nobs` number of subjects/curves.

`score.hist` a matrix of the succesive values of the scores, residual variances and log likelihood, up until convergence.

`pars` a list containing all the estimated parameters: $\boldsymbol{\lambda}_0$, $\boldsymbol{\Lambda}$, $\boldsymbol{\alpha}_k$, $\boldsymbol{\Gamma}_k$ (or $\boldsymbol{\Delta}_k$ in presence of the covariates), π_k (probabilities of cluster belongnings), σ^2, σ_x^2 (residual variance for the covariates if present), $\mathbf{v}_k$ (mean values of the covariates for each cluster).

`vars` a list containing results from the E step of the algorithm: the posterior probabilities for each subject $\pi_{k|i}$'s, the expected values of the $\boldsymbol{\gamma}_i$'s, $\boldsymbol{\gamma}_i\boldsymbol{\gamma}_i^T$, and the covariance matrix of $\boldsymbol{\gamma}_i$ given cluster membership and the observed values of the curve. See Arnqvist and Sjöstedt de Luna (2019) that explains these values.

`data` a list containing all the original data plus re-arranged functional data and covariates (if supplied).

`design` a list of spline basis matrices with and without covariates: `FullS.bmat` is the spline basis matrix $\mathbf{S}$ computed on the grid of uniquily specified time points; `FullS` is the spline basis matrix `FullS.bmat` or $\mathbf{U}$ matrix from the svd of `FullS` (if applied); $\mathbf{S}$ is the spline basis matrix computed on `timeindex`, a vector of time indices from T possible from `grid`; the inverse $(\mathbf{S}^T\mathbf{S})^{-1}$; `tag.S` is the matrix $\mathbf{S}$ with covariates; `tag.FullS` is the matrix `FullS` with covariates. See Arnqvist and Sjöstedt de Luna (2019) for further details.

`initials` a list of initial settings: q is the spline basis dimension, N is the number of subjects/curves, Q is the number of basis dimension plus the number of covariates (if present), $random$ is whether k-means was used to initialize cluster belonings, h is the vector dimension in low-dimensionality representation of the curves, K is the number of clusters, r is the number of scalar covariates, moc TRUE/FALSE signaling if the model includes covariates.

Author(s)

Per Arnqvist, Natalya Pya Arnqvist, Sara Sjöstedt de Luna

References

Arnqvist, P., Bigler, C., Renberg, I., Sjöstedt de Luna, S. (2016). Functional clustering of varved lake sediment to reconstruct past seasonal climate. *Environmental and Ecological Statistics*, **23**(4), 513–529.

Abramowicz, K., Arnqvist, P., Secchi, P., Sjöstedt de Luna, S., Vantini, S., Vitelli, V. (2017). Clustering misaligned dependent curves applied to varved lake sediment for climate reconstruction. *Stochastic Environmental Research and Risk Assessment*. Volume **31.1**, 71–85.

Arnqvist, P., and Sjöstedt de Luna, S. (2019). Model based functional clustering of varved lake sediments. *arXiv preprint arXiv:1904.10265*.

James, G.M., Sugar, C.A. (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98.462, 397–408.

See Also

`fdaMocca-package`

Examples

```
## example with lake sediment data from lake Kassjön...
library(fdaMocca)
data(varve) ## reduced data set

## run without covariates...
m <- mocca(data=varve,K=3,n.cores=2)
m
## some summary information...
summary(m)
criteria.mocca(m)
AIC(m)
BIC(m)
## various plots...
plot(m)
plot(m,select=2)
plot(m,type=2,years=c(-750:750))
plot(m,type=2,probs=TRUE,pts=TRUE,years=c(-750:750))
plot(m,type=2,pts=TRUE,select=c(3,1),years=c(-750:750))
plot(m,type=3)
plot(m,type=3,covariance=FALSE)


## model with two covariates...
## note, it takes some time to analyze the data...
m1 <- mocca(data=varve, use.covariates=TRUE,index.cov=c(2,3), K=3,n.cores=2)
m1
## summary information...
summary(m1)
criteria.mocca(m1)
## various plots...
plot(m1)
plot(m1,type=2,pts=TRUE,years=c(-750:750))
plot(m1,type=3)
plot(m1,type=3,covariance=FALSE)
plot(m1,type=3,covariates=TRUE)

## simple simulated data...
data(simdata)
set.seed(2)
m2 <- mocca(data=simdata,K=2,q=8,h=1,lambda=1e-10,n.cores=2,EMstep.tol=1e-3)
summary(m2)
criteria.mocca(m2)
plot(m2)
plot(m2,select=2)


## even simpler simulated data
##(reduced from 'simdata', EMstep.tol set high, q lower to allow automatic testing)...
library(fdaMocca)
data(simdata0)
```

```
set.seed(2)
m3 <- mocca(data=simdata0,K=2,q=5,h=1,lambda=1e-10,n.cores=2,EMstep.tol=.5,
      EMplot=FALSE,B=simdata0$B)
summary(m3)
#plot(m3)
#plot(m3,select=2))
```

---

plot.mocca *mocca plotting*

Description

The function takes a mocca object produced by mocca() and creates cluster means plots or covariance structure within each cluster.

Usage

```
## S3 method for class 'mocca'
plot(x,type=1, select =NULL,transform=FALSE,covariance=TRUE,
    covariates =FALSE,lwd=2,ylab="",xlab="",main="",ylim=NULL,
    ncolors=NULL,probs=FALSE,pts=FALSE,size=50,
    years=NULL, years.names=NULL, ...)
```

Arguments

x	a mocca object as produced by mocca().
type	determines what type of plots to print. For type=1 (default) cluster mean curves are shown in one plot on one page together with the overall mean curve; type=2 produces the trend of the frequencies of the different clusters, together with mean probabilites (if probs=TRUE), the mean value of the included covariates (if present) within each cluster (not the model estimated covariate values) are also shown, if pts=TRUE points of the frequency trend are plotted, cluster means are shown on separate plots; type=3 illustrates the covariance (or correlation) structure within each cluster. type=2 is used with annual data.
select	allows the plot for a single cluster mean to be selected for printing with type=1 or type=2. it can also be the order of the cluster means to be printed. If NULL (default), the cluster mean curves are in {1,2,...,K} order, where K is the number of clusters. If you just want the plot for the cluster mean of the second cluster set select=2.
transform	logical, informs whether svd back-transformation of the spline model matrix should be applied (see Arnqvist and Sjöstedt de Luna, 2019).
covariance	logical, informs whether covariance (TRUE) or correlation (FALSE) matrices should be plotted
covariates	logical, informs whether covariates should be added when printing the covariance structure of the spline coefficients

`lwd`	defines the line width.
`ylab`	If supplied then this will be used as the `y` label for all plots.
`xlab`	If supplied then this will be used as the `x` label for all plots.
`main`	Used as title for plots if supplied.
`ylim`	If supplied then this pair of numbers are used as the `y` limits for each plot. Default `ylim=c(-45,55)`.
`ncolors`	defines the number of colors (≥ 1) to be in the palette, used with the `rainbow()` function. If `NULL` (default), `ncolors` equals the number of clusters `K`.
`probs`	logical, used with `type=2`, informs whether the mean probabilites should be printed.
`pts`	logical, used with `type=2`, if `TRUE` (default) points of the frequency trend are shown.
`size`	the bin size used with `type=2` (default: 50 years), the bin size of how many of those years belong to a specific cluster.
`years`	a vector of years used with annual data and needed for `type=2` plot to calculate frequencies in the bins of size provided by the `size` argument.
`years.names`	a character vector that gives names of the years needed for `type=2` plot. This can be also supplied with `data`. With `varve` data `years.names` are supplied as rownames of the matrix of covariates. if `years.names=NULL` (default) then `years` are converted to the character vector and used as `years.names`.
`...`	other graphics parameters to pass on to plotting commands.

Value

The function generates plots.

Author(s)

Per Arnqvist, Sara Sjöstedt de Luna, Natalya Pya Arnqvist

References

Arnqvist, P., and Sjöstedt de Luna, S. (2019). Model based functional clustering of varved lake sediments. *arXiv preprint arXiv:1904.10265*.

See Also

`mocca`

Examples

```
## see ?mocca help files
```

print.mocca *Print a mocca object*

Description

The default print method for a mocca object.

Usage

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

Arguments

x, ...	fitted model objects of class mocca as produced by mocca().

Details

Prints out whether the model is fitted with or without covariates, the number of clusters, the estimated residual variance for the functional data and for the scalar covariates (if present), the number of covariates (if present), the maximized value of the log likelihood, and the number of subjects/curves.

Value

No return value, the function prints out several fitted results.

Author(s)

Per Arnqvist, Natalya Pya Arnqvist, Sara Sjöstedt de Luna

simdata *Simulated data*

Description

simdata is a simple test data set simulated from two clusters and consisting of 100 curves, with 50 curves belonging to one cluster and 50 to another. The test data set is a copy of the simdata of James and Sugar (2003).

Format

simdata is a list of three vectors called as x, curve and time. simdata0 is a reduced dataset with only six curves in each cluster.

Source

The data are from James and Sugar (2003).

References

James, G.M., Sugar, C.A. (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98.462, 397–408.

`summary.mocca` *Summary for a mocca fit*

Description

Takes a `mocca` object produced by `mocca()` and produces various useful summaries from it.

Usage

```
## S3 method for class 'mocca'
summary(object,...)

## S3 method for class 'summary.mocca'
print(x,digits = max(3, getOption("digits") - 3),...)
```

Arguments

`object`	a fitted `mocca` object as produced by `mocca()`.
`x`	a `summary.mocca` object produced by `summary.mocca()`.
`digits`	controls the number of digits printed in the output.
`...`	other arguments.

Value

`summary.mocca` produces the following list of summary information for a `mocca` object.

`N`	number of observations
`K`	number of clusters
`r`	number of scalar covariates if model with covariates
`sig2`	residual variance estimate for the functional data and for the scalar covariates (if the model is with covariates)
`p`	total number of the estimated parameters in the model
`tab_numOfCurves_cluster`	number of curves in each cluster as a table
`covariates_est`	mean value estimates for scalar covariates given cluster belongings (if the model is with covariates)
`t.probs`	estimated probabilities of belonging to each cluster
`crita`	a table with the maximized log likelihood, AIC, BIC and Shannon entropy values of the fitted model

Author(s)

Per Arnqvist, Sara Sjöstedt de Luna, Natalya Pya Arnqvist

See Also

mocca

Examples

```
## see ?mocca help files
```

---

varve *Varved sediment data from lake Kassjön*

Description

Annually varved lake sediment data from the lake Kassjön in Northern Sweden. The Kassjön data are used to illustrate the ideas of the model-based functional clustering with covariates. The varved sediment of lake Kassjön covers approximately 6400 years and is believed to be historical records of weather and climate. The varves (years) are clustered into similar groups based on their seasonal patterns (functional data) and additional covariates, all potentially carrying information on past climate/weather.

The Kassjön data has been analyzed in several papers. In Petterson et al. (2010, 1999, 1993) the sediment data was captured with image analysis and after preprocessing, the data was recorded as gray scale values with yearly deliminators, thus giving 6388 years (-4386 – 1901), or varves with 4–36 gray scale values per year. In Arnqvist et al. (2016) the shape/form of the yearly grey scale observations was modeled as curve functions and analyzed in a non-parametric functional data analysis approach. In Abramowicz et al. (2016) a Bagging Voronoi K-Medoid Alignment (BVKMA) method was proposed to group the varves into different "climates". The suggested procedure simultaneously clusters and aligns spatially dependent curves and is a nonparametric statistical method that does not rely on distributional or dependency structure assumptions.

Format

`varve` data is a list containing six objects named as `x`, `time`, `timeindex`, `curve`, `grid`, `covariates`. See mocca for explanation of these objects.

`varve_full` has $N = 6326$ observed subjects (years/varve), where for each varve we observed one function (seasonal pattern) and four covariates. `varve` is simply a reduced data set with only $N = 1493$ subjects.

Details

The varve patterns have the following origin. During spring, in connection to snow melt and spring runoff, minerogenic material is transported from the catchment area into the lake through four small streams, which gives rise to a bright colored layer, giving high gray-scale values (Petterson et al., 2010). During summer, autochthonous organic matter, sinks to the bottom and creates a darker layer

(lower gray-scale values). During the winter, when the lake is ice-covered, fine organic material is deposited, resulting in a thin blackish winter layer (lowest gray-scale values). There is substantial within- and between year variation, reflecting the balance between minerogenic and organic material. The properties of each varve reflect, to a large extent, weather conditions and internal biological processes in the lake the year that the varve was deposited. The minerogenic input reflects the intensity of the spring run-off, which is dependent on the amount of snow accumulated during the winter, and hence the variability in past winter climate.

The data consists of $N = 6326$ (subjects) years and the n_i observations per year ranges from 4 to 37. A few years were missing, see Arnqvist et al. (2016) for details. For each year i we observe the centered seasonal pattern in terms of grey scale values (y_i)'s at n_i time points (pixels). We also register (the four covariates) the mean grey scale within each year, the varve width n_i and the minerogenic accumulation rate (mg/cm^2) corresponding to the total amount of minerogenic material per cm^2 in the varve (year) i, see Petterson et al. (2010) for details. In order to make the seasonal patterns comparable we first put them on the same time scale $[0, 1]$, such that pixel position j at year i corresponds to position $\tilde{t}_{ij} = (j-1)/(n_i-1),\ j = 1, ..., n_i,\ i = 1, ..., N$. To make the patterns more comparable (with respect to climate) they were further aligned by landmark registration, synchronizing the first spring peaks, that are directly related to the spring flood that occurs approximately the same time each year.

As in previous analysis (Arnqvist et al., 2016) the first peak of each year is aligned towards a common spring peak with an affine warping, that is, if we denote the common spring peak as M_L and the yearly spring peak as $L_i,\ i = 1, ..., N$ and let $b = M_L/L_i$, $d = (1-M_L)/(1-L_i)$ then we will have the warped time points according to $w(t_{ij}) = t_{ij}b$ for $t_{ij} < L_i$ and $w(t_{ij}) = 1+d(t_{ij}-1)$ for $t_{ij} \geq L_i$. The common spring peak and the yearly spring peaks are given in Arnqvist et al. (2016).

Focusing on the functional forms of the seasonal patterns we finally centered them within years and worked with (the centered values) $y_i(t_{ij}) - \bar{y}_i,\ j = 1, ..., n_i,\ i = 1, ..., N$ where $\bar{y}_i = \sum_{j=1}^{n_i} y_i(t_{ij})/n_i$ is the mean grey scale value of varve (year) i. In addition to the seasonal patterns we also include four covariates: i) $x_{1i} = \bar{y}_i$, the mean grey scale; ii) $x_{2i} = n_i$, the varve width (proportional to n_i); iii) x_{3i}, the minerogenic accumulation rate corresponding to the accumulated amount of minerogenic material per cm^2 in varve i; and iv) x_{4i}, the landmark which is the distance from the start of the year to the first peak, interpreted as the start of the spring, for details see (Petterson et al., 2010, and and Arnqvist et al. 2016).

`varve_full` is a full data set with $N = 6326$ years/curves spanning the time period 4486 B.C. until 1901 A.D..

`varve` is a reduced data set with $N = 1493$ years/curves covering the time period 750 BC to 750 AD.

References

Arnqvist, P., Bigler, C., Renberg, I., Sjöstedt de Luna, S. (2016). Functional clustering of varved lake sediment to reconstruct past seasonal climate. *Environmental and Ecological Statistics*, **23**(4), 513–529.

Abramowicz, K., Arnqvist, P., Secchi, P., Sjöstedt de Luna, S., Vantini, S., Vitelli, V. (2017). Clustering misaligned dependent curves applied to varved lake sediment for climate reconstruction. *Stochastic Environmental Research and Risk Assessment*. Volume **31.1**, 71–85.

Arnqvist, P., and Sjöstedt de Luna, S. (2019). Model based functional clustering of varved lake sediments. *arXiv preprint arXiv:1904.10265*.

Petterson, G., Renberg, I., Sjöstedt de Luna, S., Arnqvist, P., and Anderson, N. J. (2010). Climatic influence on the inter-annual variability of late-Holocene minerogenic sediment supply in a boreal forest catchment. *Earth surface processes and landforms*. **35**(4), 390–398.

Petterson, G., B. Odgaard, and I. Renberg (1999). Image analysis as a method to quantify sediment components. *Journal of Paleolimnology 22*. (4), 443–455.

Petterson, G., I. Renberg, P. Geladi, A. Lindberg, and F. Lindgren (1993). Spatial uniformity of sediment accumulation in varved lake sediments in Northern Sweden. *Journal of Paleolimnology*. **9**(3), 195–208.

Index