Package: pfica (via r-universe)

October 25, 2024

Version 0.1.3

Title Independent Components Analysis Techniques for Functional Data

Author Marc Vidal [aut, cre], Ana Mª Aguilera [aut, ths]

Maintainer Marc Vidal <marc.vidalbadia@ugent.be>

Description This package includes a set of tools to perform smoothed (and non-smoothed) principal/independent components analysis of functional data. Various functional pre-whitening approaches are implemented as discussed in Vidal and Aguilera (2022) "Novel whitening approaches in functional settings", <doi:10.1002/sta4.516>. Further whitening representations of functional data can be derived in terms of a few principal components, providing a powerful avenue to explore hidden structures in low dimensional settings: see Vidal, Rosso and Aguilera (2021) "Bi-smoothed functional independent component analysis for EEG artifact removal", <doi:10.3390/math9111243>.

License GPL (>= 2)

Depends R (>= 2.10), fda

Imports expm, whitening

URL https://github.com/m-vidal/pfica

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.2.0

Repository https://marcvidalbadia.r-universe.dev

RemoteUrl https://github.com/marcvidalbadia/pfica

RemoteRef HEAD

RemoteSha a8d9d17364a03d0c3a0ebe2748056c9727de3704

Contents

`ffobi` *Smoothed functional ICA*

Description

This function computes the ordinary ICA procedure from a sample represented by basis functions (Fourier, B-splines...). The estimation method is based on the use of fourth moments (kurtosis), in which it is assumed that the independent components have different kurtosis values. The proposed algorithm is an implementation of the kurtosis operator introduced in Peña et. al (2014).

Usage

```
ffobi(fdx, ncomp = fdx$basis$nbasis, eigenfPar = fdPar(fdx),
      w = c("PCA", "PCA-cor","ZCA", "ZCA-cor",
            "Varimax", "Varimax-cor", "Cholesky"),
      pr = c("fdx", "wfdx"), center = TRUE)
```

Arguments

`fdx`	a functional data object obtained from the **fda** package.
`ncomp`	number of independent components to compute.
`eigenfPar`	a functional parameter object obtained from the **fda** package.
`w`	the whitening procedure. By default `ZCA` (Mahalanobis whitening) is used.
`pr`	the functional data object to project on to the space spanned by the eigenfunctions of the kurtosis kernel function. To compute the independent components, the usual procedure is to use `wfdx`, the whitened basis expansion. Thus, if `pr` is not supplied, `wfdx` is used.
`center`	a logical value indicating whether the mean function has to be subtracted from each functional observation.

Details

This functional ICA consists of performing the multivariate ICA of a transformation of the coordinate vectors associated with a basis of functions. The algorithm also incorporates a continuous penalty in the orthonormality constraint of the kurtosis eigenfunctions.

Value

a list with the following named entries:

`ICA.eigv`	a numeric vector giving the eigenvalues of the kurtosis kernel function.
`ICA.basis`	a functional data object for the kurtosis kernel eigenfunctions.
`ICA.scores`	a matrix whose column vectors are the projection coefficients for `fdx` or `wfdx`.

Author(s)

Marc Vidal, Ana Mª Aguilera

References

Peña, C., J. Prieto and C. Rendón (2014). *Independent components techniques based on kurtosis for functional data analysis*. Working paper 14–10 Statistics and Econometric Series (06); Universidad Carlos III de Madrid, Madrid, 2014.

See Also

kffobi, whiten.fd

Examples

```
## Canadian Weather data
library(fda)
arg <- 1:365
Temp <- CanadianWeather$dailyAv[,,1]
q <- 14
B <- create.bspline.basis(rangeval=c(min(arg),max(arg)), nbasis=q)
x <- Data2fd(Temp, argvals = arg, B)
#plot(x) #plot data
Lfdobj <- int2Lfd(max(0, norder(B)-2))
penf <- fdPar(B, Lfdobj, lambda=4200) #penalty parameter (lambda)
ica.fd <- ffobi(x,  eigenfPar = penf, w = "Cholesky")

## Plot by  minimmum smoothed kurtosis
sc <- ica.fd$ICA.scores
plot(sc[,q], sc[,(q-1)], pch = 20, col = factor(c(rep("East",15),rep("West",20))),
     ylab = "IC q-1", xlab = "IC q")
```

---

kffobi *Smoothed functional PCA/ICA*

Description

This function computes the ordinary ICA procedure on a smoothed principal component expansion (also known as Karhunen-Loeve expansion) whose eigenbasis is expressed in terms of basis functions (Fourier, B-splines...). The estimation method is based on the use of fourth moments (kurtosis), in which it is assumed that the independent components have different kurtosis values. The function further provides smoothed (and non-smoothed) functional PCA estimates based on Ocaña et al. (1999).

Usage

```
kffobi(fdx, ncomp = fdx$basis$nbasis, eigenfPar = fdPar(fdx),
       w = c("PCA", "PCA-cor","ZCA", "ZCA-cor","Cholesky"),
       pr = c("fdx", "wfdx", "KL", "wKL"),
       center = TRUE)
```

Arguments

fdx	a functional data object obtained from the **fda** package.
ncomp	number of independent components to compute (must be > 1).
eigenfPar	a functional parameter object obtained from the **fda** package.
w	the whitening procedure. By default ZCA (Mahalanobis whitening) is used.
pr	the functional data object to project on to the space spanned by the eigenfunctions of the kurtosis kernel function. To compute the independent components, the usual procedure is to use wKL, the whitened principal component expansion. If pr is not supplied, then wKL is used. Note that w stands for whitened, e.g., wfdx is the whitened original data.
center	a logical value indicating whether the mean function has to be subtracted from each functional observation.

Details

This functional ICA consists of performing the multivariate ICA of the PC coordinate vectors in terms of the principal component weight functions.

Value

a list with the following named entries:

PCA.eigv	a numeric vector giving the eigenvalues of the covariance kernel function.
PCA.basis	a functional data object for the eigenfunctions of the covariance kernel function.
PCA.scores	a matrix whose column vectors are the principal components.
ICA.eigv	a numeric vector giving the eigenvalues of the kurtosis kernel function.
ICA.eigv	a numeric vector giving the eigenvalues of the kurtosis kernel function.
ICA.scores	a matrix whose column vectors are the projection coefficients for fdx, wfdx, KL or wKL.
wKL	the whitened principal components expansion whith coefficients in terms of basis functions.

Author(s)

Marc Vidal, Ana Mª Aguilera

References

Li, B., G. Van Bever, H. Oja, R. Sabolova and F. Critchley (2019). *Functional independent component analysis: an extension of the fourth-orderblind identification*. Technical Report, Univ. Namur.

Ocaña, F.A., A.M. Aguilera and M.J. Valderrama (1999). *Functional Principal Components Analysis by Choice of Norm*. Journal of Multivariate Analysis, 71(2), <doi:10.1006/jmva.1999.1844>.

See Also

ffobi, pspline.kffobi, whiten.fd

Examples

```
## Canadian Weather data
library(fda)
arg <- 1:365
Temp <- CanadianWeather$dailyAv[,,1]
B <- create.bspline.basis(rangeval=c(min(arg),max(arg)), nbasis=14)
x <- Data2fd(Temp, argvals = arg, B)
#plot(x) #plot data
Lfdobj <- int2Lfd(max(0, norder(B)-2))
penf <- fdPar(B, Lfdobj, lambda=10^9) #penalty parameter (lambda)
ica.fd <- kffobi(x, 2, eigenfPar = penf, w="ZCA-cor")

## Whitened data using the two first smoothed principal components
wKL <- ica.fd$wKL

## Plot by region
sc <- ica.fd$ICA.scores
plot(sc[,1], sc[,2], pch = 20, col = factor(CanadianWeather$region),
     ylab = "IC 1", xlab = "IC 2")
```

`pspline.kffobi` *P-Spline smoothed functional PCA/ICA*

Description

This function is an alternative way of computing the smoothed functional ICA in terms of principal components (see `kffobi`). The function further provides smoothed (and non-smoothed) functional PCA estimates. A discrete penalty that measures the roughness of principal factors by summing squared r-order differences between adjacent B-spline coefficients (P-spline penalty) is used; see Aguilera and Aguilera-Morillo (2013) and Eliers and Marx (2021).

Usage

```
pspline.kffobi(fdx, ncomp = fdx$basis$nbasis, pp = 0, r = 2,
               w = c("PCA", "PCA-cor","ZCA", "ZCA-cor","Cholesky"),
               pr = c("fdx", "wfdx", "KL", "wKL"),
               center = TRUE)
```

Arguments

`fdx`	a functional data object obtained from the **fda** package.
`ncomp`	number of independent components to compute (must be > 1).
`pp`	the penalty parameter. It can be estimated using *leave-one-out* cross-validation.
`r`	a number indicating the order of the penalty.
`w`	the whitening procedure. By default `ZCA` (Mahalanobis whitening) is used.

pr	the functional data object to project on to the space spanned by the eigenfunctions of the kurtosis kernel function. To compute the independent components, the usual procedure is to use wKL, the whitened principal component expansion. If pr is not supplied, then wKL is used. Note that w stands for whitened, e.g., wfdx is the whitened original data.
center	a logical value indicating whether the mean function has to be subtracted from each functional observation.

Value

a list with the following named entries:

PCA.eigv	a numeric vector giving the eigenvalues of the covariance kernel function.
PCA.basis	a functional data object for the eigenfunctions of the covariance kernel function.
PCA.scores	a matrix whose column vectors are the principal components.
ICA.eigv	a numeric vector giving the eigenvalues of the kurtosis kernel function.
ICA.eigv	a numeric vector giving the eigenvalues of the kurtosis kernel function.
ICA.scores	a matrix whose column vectors are the projection coefficients for fdx, wfdx, KL or wKL.
wKL	the whitened principal components expansion whith coefficients in terms of basis functions.

Author(s)

Marc Vidal, Ana Mª Aguilera

References

Aguilera, A.M. and M.C. Aguilera-Morillo (2013). *Penalized PCA approaches for B-spline expansions of smooth functional data.* Applied Mathematics and Computation 219(14), 7805–7819, <doi:10.1016/j.amc.2013.02.009>.

Eilers, P. and B. Marx (2021). *Practical Smoothing: The Joys of P-splines.* Cambridge: Cambridge University Press, <doi:10.1017/9781108610247>.

Vidal, M., M. Rosso and A.M. Aguilera. (2021). *Bi-Smoothed Functional Independent Component Analysis for EEG Artifact Removal*. Mathematics 9(11), 1243, <doi:10.3390/math9111243>.

See Also

kffobi, whiten.fd

Examples

```
## Canadian Weather data
library(fda)
arg <- 1:365
Temp <- CanadianWeather$dailyAv[,,1]
B <- create.bspline.basis(rangeval=c(min(arg),max(arg)), nbasis=14)
x <- Data2fd(Temp, argvals = arg, B)
```

```
#plot(x) #plot data
ica.fd <- pspline.kffobi(x, 2, pp = 10^4, w="ZCA-cor")

## Whitened data using the two first smoothed principal components
wKL <- ica.fd$wKL

## Plot by region
sc <- ica.fd$ICA.scores
plot(sc[,2], sc[,1], pch = 20, col = factor(CanadianWeather$region),
     ylab = "IC 1", xlab = "IC 2")
```

`whiten.fd` *Whitening (or sphering) functional data*

Description

This function whitens functional data in terms of basis functions expansions.

Functional whitening procedures available:

- `PCA`: uses a projection that combines the covariance function eigenfactors and an arbitrary orthonormal basis
- `PCA-cor`: PCA whitening based on the correlation function
- `ZCA`: or Mahalanobis whitening, is a symmetric solution based on the spectral decomposition of the inverse square root of the covariance function
- `ZCA-cor`: Mahalanobis whitening based on the correlation function
- `Varimax`: combines the covariance function eigenfactors and its varimax rotation
- `Varimax-cor`: Varimax whitening based on the correlation function
- `Cholesky`: uses the Cholesky factorization of the inverse covariance function.

Which functional whitening procedure to use? `PCA` and `PCA-cor` allow for maximum compression of the functional observations and therefore the new data representations might be rather noisy. `ZCA` and `ZCA-cor` maximize the similarity with the original data, providing most robust whitening. The `Varimax`/`Varimax-cor` procedures, which are analogous to `ZCA`/`ZCA-cor`, can be used when the first component of the functional PCA accounts for a high percentage of the total variance. `Cholesky` takes an intermediate position in scoring for maximum compression and similarity indices. To explore low dimensional structures in the data, `kffobi` and `pspline.kffobi` allows to compute these functional whitening representations using a few principal components.

For further details, see Vidal and Aguilera (2022).

Usage

```
whiten.fd(fdx, w = c("PCA", "PCA-cor","ZCA", "ZCA-cor",
                     "Varimax", "Varimax-cor", "Cholesky"),
          pre.center = TRUE, post.center = FALSE)
```

Arguments

fdx	a functional data object obtained from the **fda** package.
w	the whitening procedure. By default ZCA (Mahalanobis whitening) is used.
pre.center	centering the functional observations before performing the transformation.
post.center	centering the functional observations after performing the transformation.

Details

As in the **whitening** package (see Kessy et al. (2018)), to solve the sign ambiguity in PCA, PCA-cor we use eigenvector matrices with a positive diagonal. The pre.centrer/post.centrer options can be used interchangeably.

Value

wfdx	a functional data object with whitened coefficients.

Author(s)

Marc Vidal, Ana Mª Aguilera

References

Acal, C., A.M. Aguilera and M. Escabias (2020). *New Modeling Approaches Based on Varimax Rotation of Functional Principal Components*. Mathematics, 8(11), 1-15. <doi:10.3390/math8112085>

Kessy, A., A. Lewin and K. Strimmer (2018). *Optimal Whitening and Decorrelation*. The American Statistician, 72(4), 309-314, <doi:10.1080/00031305.2016.1277159>.

Vidal, M. and and A.M. Aguilera (2022). *Novel whitening approaches in functional settings*. Stat, e516, <doi:10.1002/sta4.516>.

See Also

ffobi

Examples

```
## Canadian Weather data
library(fda)
arg <- 1:365
Temp <- CanadianWeather$dailyAv[,,1]
B <- create.bspline.basis(rangeval=c(min(arg),max(arg)), nbasis=16)
x <- Data2fd(Temp, argvals = arg, B)
wx <- whiten.fd(x)
```

Index