Package ‘fregion’

October 3, 2016

Type Package

Title Confidence Regions and Bands for Functional Data

Version 0.091

Date 2016-10-03

Description This package provides inferential tools for functional parameters when Gaussian estimators are in place.

License GPL-3

Imports CompQuadForm,
fda

LazyData TRUE

RoxygenNote 5.0.1

R topics documented:

`Data2bifd` *Creates a bifd object from a matrix.*

Description

Creates a bifd object from a matrix.

Usage

```
Data2bifd(sargvals = NULL, targvals = NULL, y = NULL, sbasisobj = NULL,
  tbasisobj = NULL, Lfdobjs = NULL, Lfdobjt = NULL, lambdas = NULL,
  lambdat = NULL)
```

Arguments

`sargvals`	Argument values for ‘s’ in the matrix input y(s,t).
`targvals`	Argument values for ‘t’ in the matrix input y(s,t).
`y`	Data (only one) as a matrix.
`sbasisobj`	basisfd object for ‘s’
`tbasisobj`	basisfd object for ‘t’
`Lfdobjs`	Number of derivatives used for smoothing along ‘s’. No smoohting will be done for NULL
`Lfdobjt`	Number of derivatives used for smoothing along ‘t’. No smoohting will be done for NULL
`lambdas`	Weight on the smoothing along ‘s’
`lambdat`	Weight on the smoothing along ‘t’

`eigen.fd` *Takes eigen decomposition of bifd covariance object.*

Description

Takes eigen decomposition of bifd covariance object.

Usage

```
eigen.fd(cov.fd)
```

Arguments

`cov.fd`	Covariance operator as a bifd object. Use Data2bifd to convert covariance matrix into a bifd object.

fregion *fregion*

Description

This package contains example codes for inferential tools suggested in http://arxiv.org/abs/1607.07771, titled 'Geometric Approach to Confidence Regions and Bands for Functional Parameters'. The top level functions below take functional estimate and the covariance of it to perform hypothesis testings, to construct confidence bands, and to construct/visuallize hyper-rectangular regions. The inputs can be either vector/matrix or fd/bifd object from fda package.

- fregion.test: Perform hypothesis testings based on confidence regions.
- fregion.band: Construct (simultaneous) confidence bands using hyper-ellipsoid confidence regions.
- fregion.rect: Construct a hyper-rectangular confidence region in Hilbert space. It can be visuzlized using plot.fregion.rect.

The example below is given in a mean function estimation context but works for other functional estimates as long as the estimators are Gaussian.

Examples

```
# 1. Vector/matrix version

# Generate a sample
p = 200 ; N = 80 ; rangeval = c(0,1)
grid = make.grid(p, rangevals=rangeval)
mu0 = meanf.poly(grid,c(0,1)) ; names(mu0) = grid
mu = meanf.poly(grid,c(0,1.1)) ; names(mu) = grid
cov.m = make.cov.m(cov.f = covf.st.matern, grid=grid, cov.f.params=c(2/2,1,1))
x = make.sample(mu,cov.m,N)

# Find the estimate and covariance
hat.mu = rowMeans(x)
hat.cov.m = crossprod(t(x - hat.mu)) / (N-1)
e.hat.cov.m = eigen(hat.cov.m)   # <- This is optional and can be provide into the functions instead of hat.cov.m

# Compare different methods for Hypothesis testings.
(a1 <- fregion.test(hat.mu,mu0,hat.cov.m,N,type=c("ALL"),pc.cut=c(1,3,4,5,0.99,0.999)))

# Make and visualize/compare confidence bands
b <- fregion.band(hat.mu,hat.cov.m,N=N,type=c("Bs","BEc","BEPC.10","naive.t"),conf.level=c(0.95))
plot(b)

# Make rectangular region and visulize
c <- fregion.rect(hat.mu-mu0,hat.cov.m,N=N)
plot(c)
```

```
# 2. fd/bifd version

# create basis, convert vector/matrix into fd/bifd objects.
require(fda)
nbasis <- round(p*0.9)
fd.basis <- create.bspline.basis(rangeval,nbasis)
mu0.fd <- Data2fd(names(mu0),mu0,fd.basis)
mu.fd <- Data2fd(names(mu),mu,fd.basis)
x.fd <- Data2fd(rownames(x),x,fd.basis)
hat.mu.fd <- mean.fd(x.fd)
hat.cov.fd <- var.fd(x.fd)
e.hat.cov.fd <- eigen.fd(hat.cov.fd)   # <- This is optional and can be provide into the functions instead of hat.

# Compare different methods for Hypothesis testings.
(a1.fd <- fregion.test(hat.mu.fd, mu0.fd, hat.cov.fd, N, type=c("ALL"), pc.cut=c(1,3,4,5,0.99,0.999)))

# Make and visualize/compare confidence bands
b.fd <- fregion.band(hat.mu.fd,hat.cov.fd,N=N,type=c("Bs","BEc","BEPC.10","naive.t"),conf.level=c(0.95))
plot(b.fd)

# Make rectangular region and visulize
c.fd <- fregion.rect(hat.mu.fd-mu0.fd,hat.cov.fd,N=N)
plot(c.fd)
```

fregion.band *Makes confidence bands*

Description

Makes confidence bands

Usage

```
fregion.band(x, cov, N = 1, type = c("Bs", "BEc"), conf.level = c(0.95),
  grid.size = 200, inv.method = "gamma.approx", prec = NULL,
  sim.size = 10000)
```

Arguments

x	Functional parameter estimate. It can be either a vector or fd object from fda.
cov	N * Cov(X), in which X is the functional estimator. It can be either matrix or bifd object from fda. The eigen decomposition of Cov(X) can be used instead.
N	It should be '1' if 'cov' is the covariance operator for X itself, which is the default value.
type	The band(s) to be constructed. • BEc : The suggested modified Scheffe style band from hyper-ellipsoie Ec, which uses up to the very last dimension.

- BEc1 : Another modified Scheffe style band from hyper-ellipsoie Ec1. (Wider than BEc although it does not require smoothness assumption. For comparision purpose only, not recommend for use)
- Bs : Parametric bootstrap simultaneous confidence band, similar to the one appeard in Degras(2011) (for comparision purpose)
- naive-t : A collection of point-wise t-intervals. (for comparision purpose)
- BEPC.x : Scheffe band from 'x' dimensional chi-square ellipse. (for comparision purpose)

`conf.level`	A vector of confidence levels for the bands to achieve.
`grid.size`	This determines on how fine grid the bands will be constructed before converted as an 'fd' object. This parameter is used only when 'x' is fd object and 'cov' is bifd object.
`inv.method`	(Currently Not Used) This determines how the inverse of the sum of chi squares distribution will be achieved. It can be either "gamma.approx", or "inv.imhof". Currently, only "gamma.approx" works.
`prec`	(Currently Not Used) This determines the accuracy of imhof. It's used only when inv.method is inv.imhof.
`sim.size`	This determines bootstrap sample size for Bs

Value

fregion.band Either a collection of vector valued bands or 'fd' object whose objectname is changed to fregion.band.

Examples

```
# 1. Vector/matrix version

# Generate a sample
p = 200 ; N = 80 ; rangeval = c(0,1)
grid = make.grid(p, rangevals=rangeval)
mu0 = meanf.poly(grid,c(0,1)) ; names(mu0) = grid
mu = meanf.poly(grid,c(0,1.1)) ; names(mu) = grid
cov.m = make.cov.m(cov.f = covf.st.matern, grid=grid, cov.f.params=c(2/2,1,1))
x = make.sample(mu,cov.m,N)

# Find the estimate and covariance
hat.mu = rowMeans(x)
hat.cov.m = crossprod(t(x - hat.mu)) / (N-1)
e.hat.cov.m = eigen(hat.cov.m)   # <- This is optional and can be provide into the functions instead of hat.cov.m

# Make and visualize/compare confidence bands
b <- fregion.band(hat.mu,hat.cov.m,N=N,type=c("Bs","BEc","BEPC.10","naive.t"),conf.level=c(0.95))
plot(b)


# 2. fd/bifd version

# create basis, convert vector/matrix into fd/bifd objects.
```

```
require(fda)
nbasis <- round(p*0.9)
fd.basis <- create.bspline.basis(rangeval,nbasis)
mu0.fd <- Data2fd(names(mu0),mu0,fd.basis)
mu.fd <- Data2fd(names(mu),mu,fd.basis)
x.fd <- Data2fd(rownames(x),x,fd.basis)
hat.mu.fd <- mean.fd(x.fd)
hat.cov.fd <- var.fd(x.fd)
e.hat.cov.fd <- eigen.fd(hat.cov.fd)   # <- This is optional and can be provide into the functions instead of hat.

# Make and visualize/compare confidence bands.
b.fd <- fregion.band(hat.mu.fd,hat.cov.fd,N=N,type=c("Bs","BEc"), conf.level=c(0.95, 0.9))
plot(b.fd)
```

`fregion.rect` *Builds a Rectangular Confidence Region*

Description

Builds a Rectangular Confidence Region

Usage

```
fregion.rect(x, cov, N = 1, type = "Rz", conf.level = 0.95,
  pc.cut = 0.999, df = NULL)
```

Arguments

`x`	Functional parameter estimate. It can be either a vector or fd object from fda.
`cov`	N * Cov(X), in which X is the functional estimator. It can be either matrix or bifd object from fda. The eigen decomposition of Cov(X) can be used instead.
`N`	It should be '1' if 'cov' is the covariance operator for X itself, which is the default value.
`type`	This specifies which rectangular region to be constructed. It should be either one of "Rz", "Rz1", "Rzs", or "Rz1s".
`conf.level`	Confidence level of the region.
`pc.cut`	A numeric value. For integer values, fPC up to those values will be used. If it's a value from 0 to 1, this specifies the proportion of (estimated) variance that should be explained by the fPCs. If it is 0, all the available fPCs will be used as long as the size of eigenvalues are greater than .Machine$double.eps.
`df`	Degree of freedom to use in small sample versions.

Value

fregion.rect Use plot.fregion.rect to visualize.

Examples

```
# 1. Vector/matrix version

# Generate a sample
p = 200 ; N = 80 ; rangeval = c(0,1)
grid = make.grid(p, rangevals=rangeval)
mu0 = meanf.poly(grid,c(0,1)) ; names(mu0) = grid
mu = meanf.poly(grid,c(0,1.1)) ; names(mu) = grid
cov.m = make.cov.m(cov.f = covf.st.matern, grid=grid, cov.f.params=c(2/2,1,1))
x = make.sample(mu,cov.m,N)

# Find the estimate and covariance
hat.mu = rowMeans(x)
hat.cov.m = crossprod(t(x - hat.mu)) / (N-1)
e.hat.cov.m = eigen(hat.cov.m)   # <- This is optional and can be provide into the functions instead of hat.cov.m

# Make rectangular region and visulize
c <- fregion.rect(hat.mu-mu0,hat.cov.m,N=N)
plot(c)


# 2. fd/bifd version

# create basis, convert vector/matrix into fd/bifd objects.
require(fda)
nbasis <- round(p*0.9)
fd.basis <- create.bspline.basis(rangeval,nbasis)
mu0.fd <- Data2fd(names(mu0),mu0,fd.basis)
mu.fd <- Data2fd(names(mu),mu,fd.basis)
x.fd <- Data2fd(rownames(x),x,fd.basis)
hat.mu.fd <- mean.fd(x.fd)
hat.cov.fd <- var.fd(x.fd)
e.hat.cov.fd <- eigen.fd(hat.cov.fd)   # <- This is optional and can be provide into the functions instead of hat.

# Make rectangular region and visulize
c.fd <- fregion.rect(hat.mu.fd-mu0.fd,hat.cov.fd,N=N)
plot(c.fd)
```

`fregion.test` *Performs hypothesis test(s) based on functional confidence region(s).*

Description

Performs hypothesis test(s) based on functional confidence region(s).

Usage

```
fregion.test(x, x0 = 0, cov, N = 1, type = c("Ec"), pc.cut = c(0.99),
  prec = NULL, hat.cov = NULL, df = NULL)
```

Arguments

`x`	Functional parameter estimate. It can be either a vector or fd object from fda.
`x0`	Functional parameter under the null hypothesis. Zero function is assumed if it's not given.
`cov`	N * Cov(X), in which X is the functional estimator. It can be either matrix or bifd object from fda. The eigen decomposition of Cov(X) can be used instead.
`N`	It should be '1' if 'cov' is the covariance operator for X itself, which is the default value.
`type`	This specifies which regions to be used for the tests. It should be a collection of the following: "Enorm", "Epc", "Ec", "Ec1", "Rz", "Rz1", "Rzs", or "Rz1s". • Enorm : The Hilbert space norm based test, which uses a 'ball' in the function space(H). • Epc : fPCA based test, which is a finite-dimensional chi-square ellipse in H • Ec : The suggested test based on the ellipsoid region in which radius are proportional to the square-root of corresponding eigenvalues. • Ec1 : The second suggestion for ellipsoie region, in which radius are proportional to the square-root of tails sums of eigenvalues. Doesn't require any smoothness assumption. • Rz : The suggested rectangular region based test. • Rz1 : The second suggested rectangular region based test. • Rzs : The small sample version of Rz. • Rz1s : The small sample version of Rz1.
`pc.cut`	It takes a vector of number of fPC to use in each HT. For integer values, fPC up to those values will be used. If it's a value from 0 to 1, this specifies the proportion of (estimated) variance that should be explained by the fPCs. If it is 0, all the available fPCs will be used as long as the size of eigenvalues are greater than .Machine$double.eps.
`prec`	This determines the accuracy of imhof. One may try to modify this if p-value achieved in Ellipsoid form other than Epc gives negative value. It should the the form of c(epsabs, epsrel, limit).
`hat.cov`	An optional estimated covariance operator, which will be used when 'cov' is given as the true covariance and small sample version(s) are used. Usually not needed.
`df`	Degree of freedom to use in small sample versions.

Examples

```
# 1. Vector/matrix version

# Generate a sample
p = 200 ; N = 80 ; rangeval = c(0,1)
grid = make.grid(p, rangevals=rangeval)
mu0 = meanf.poly(grid,c(0,1)) ; names(mu0) = grid
mu = meanf.poly(grid,c(0,1.2)) ; names(mu) = grid
cov.m = make.cov.m(cov.f = covf.st.matern, grid=grid, cov.f.params=c(2/2,1,1))
```

```
x = make.sample(mu,cov.m,N)

# Find the estimate and covariance
hat.mu = rowMeans(x)
hat.cov.m = crossprod(t(x - hat.mu)) / (N-1)
e.hat.cov.m = eigen(hat.cov.m)   # <- This is optional and can be provide into the functions instead of hat.cov.m

# Compare different methods for Hypothesis testings.
(a1 <- fregion.test(hat.mu,mu0,hat.cov.m,N,type=c("ALL"),pc.cut=c(1,3,4,5,0.99,0.999)))


# 2. fd/bifd version

# create basis, convert vector/matrix into fd/bifd objects.
require(fda)
nbasis <- round(p*0.9)
fd.basis <- create.bspline.basis(rangeval,nbasis)
mu0.fd <- Data2fd(names(mu0),mu0,fd.basis)
mu.fd <- Data2fd(names(mu),mu,fd.basis)
x.fd <- Data2fd(rownames(x),x,fd.basis)
hat.mu.fd <- mean.fd(x.fd)
hat.cov.fd <- var.fd(x.fd)
e.hat.cov.fd <- eigen.fd(hat.cov.fd)   # <- This is optional and can be provide into the functions instead of hat.

# Compare different methods for Hypothesis testings.
(a1.fd=fregion.test(hat.mu.fd,mu0.fd,hat.cov.fd,N,type=c("ALL"),pc.cut=c(1,3,4,5,0.99,0.999)))
```

plot.fregion.band *Visualizes confidence bands constructed from fregion.band.*

Description

Visualizes confidence bands constructed from fregion.band.

Usage

```
## S3 method for class 'fregion.band'
plot(band, center = TRUE, legendx = "topleft",
  legendy = NULL, ...)
```

Arguments

band	A 'fregion.band' object, the output from fregion.band funciton.
center	Whether to include the functional estimate or not.
legendx	position 'x' of the legend. If NULL is passed, the legend will not be drawn (However, it may be added manually)
legendy	position 'y' of the legend.

`...`	Graphical parameters to be passed/overrided. If 'center' is TRUE, the first elements of 'col', 'lwd', 'lty' will be used for the estimate and the next ones will be used for the bands, but using the same values for one pair, i.e., lower and upper bounds.

`plot.fregion.rect` *Visualizes rectangular confidence regions achieved from fregion.rect.*

Description

Visualizes rectangular confidence regions achieved from fregion.rect.

Usage

```
## S3 method for class 'fregion.rect'
plot(rect, npc = NULL, returntype = NULL,
  main = NULL, xlab = NULL, ylab = NULL, grid.size = 200)
```

Arguments

`rect`	A 'fregion.rect' object, the output from fregion.rect funciton.
`npc`	Number of fPCs to plot.
`grid.size`	Since this function makes plots, fd objects are evaluated and gird.size determines how find the grid should be.

Index