Package ‘ImageSCC’

May 23, 2019

Type Package

Title SCC for Mean Function of Imaging Data

Version 0.1.0

Author Yueying Wang, Guannan Wang and Li Wang

Maintainer Yueying Wang <yueyingw@iastate.edu> and Guannan Wang <gwang01@wm.edu>

Depends R (>= 2.10), MASS, Matrix, BPST, Triangulation, graphics, RSpectra, pracma

Description This R package is the implementation program for manuscript entitled ``Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data'' by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

basis.image	*Basis genration of all the triangulation*

Description

Basis genration of all the triangulation

Usage

```
basis.image(Z, V.ests, Tr.ests, d.est, r)
```

boot.image	*Triangulation selection for contructing the SCC via bootstrap*

Description

The function selects triangulation used for constructing SCC by bootstrap method.

Usage

```
boot.image(Ya, Yb = NULL, Z, d.est, d.band, r, V.est.a, Tr.est.a,
  V.est.b = NULL, Tr.est.b = NULL, V.bands, Tr.bands, lambda,
  nboot = 50, alpha0 = 0.05, adjust.sigma = TRUE)
```

Arguments

`Ya`	a matrix of data with each row corresponding to one subject/image.
`Yb`	an optional matrix containing the second group of imaging data. When `Yb` is NULL, triangulation selection focuses on one sample SCC, otherwise it focuses on the SCC for mean difference ($\mu_2 - \mu_1$) between two sets of images.
`Z`	a 2-column matrix specifying locations of information.
`d.est`	degree of bivariate spline for mean estimation.
`d.band`	degree of bivariate spline for SCC.
`r`	smoothness parameter.
`V.est.a`	the 2-column matrix of vertices' coordinates in the triangulation for estimating mean in the first sample.
`Tr.est.a`	the 3-column matrix specifying triangles in the triangulation. Each row contains 3 indices of vertices corresponding to one triangle in the triangulation.
`V.est.b, Tr.est.b`	optional information of triangulation used for estimating mean in the second sample.
`V.bands, Tr.bands`	lists of candidates for triangulations used to construct SCC.
`lambda`	the vector of the candidates for penalty parameter when estimating mean function.
`nboot`	number of bootstrap iterations. Default is 50.
`alpha0`	a value specifying confidence level of SCC.
`adjust.sigma`	a logical value indicating whether $\sigma(z)$ is adjusted when constructing SCC. Default is TRUE.

Details

This R package is the implementation program for manuscript entitled "Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data" by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

Examples

```
# Triangulation information;
data(Brain.V1); data(Brain.Tr1); # triangulation No. 1;
data(Brain.V2); data(Brain.Tr2); # triangulation No. 2;
data(Brain.V3); data(Brain.Tr3); # triangulation No. 3;
#' V.est=Brain.V2; Tr.est=Brain.Tr2;
V.bands=list(V1=Brain.V1,V2=Brain.V2,V3=Brain.V3);
Tr.bands=list(Tr1=Brain.Tr1,Tr2=Brain.Tr2,Tr3=Brain.Tr3);
# Location information;
n1=40; n2=40;
npix=n1*n2
u1=seq(0,1,length.out=n1)
v1=seq(0,1,length.out=n2)
uu=rep(u1,each=n2)
vv=rep(v1,times=n1)
Z=as.matrix(cbind(uu,vv))
ind.inside=inVT(V.est,Tr.est,Z[,1],Z[,2])$ind.inside
# Parameters for bivariate spline over triangulation;
d.est=5; d.band=2; r=1;

# Example 1. One-group SCC;
# simulation parameters
n=50; lam1=0.5; lam2=0.2; mu.func=2; noise.type='Func';
lambda=10^{seq(-6,3,0.5)}; alpha.grid=c(0.1,0.05,0.01);
dat=data1g.image(n,Z,ind.inside,mu.func,noise.type,lam1,lam2)
Y=dat$Y; beta.true=dat$beta.true;
tri.band=boot.image(Ya=Y,Z=Z,d.est=d.est,d.band=d.band,r=r,V.est.a=V.est,Tr.est.a=Tr.est,V.bands=V.bands,Tr
tri.band$tri.band
V.band=tri.band$V.band.a; Tr.band=tri.band$Tr.band.a;

# Example 2. Two-group SCC;
# simulation parameters
na=50; nb=60; lam1=0.5; lam2=0.2; mu1.func=1; delta=0.3;
noise.type='Func'; lambda=10^{seq(-6,3,0.5)}; alpha.grid=c(0.10,0.05,0.01);
dat=data2g.image(na,nb,Z,ind.inside,mu1.func,noise.type,lam1,lam2,delta)
Ya=dat$Ya; Yb=dat$Yb; beta.true=dat$beta.true;
beta.diff=beta.true[,2]-beta.true[,1]
V.est.a=V.est.b=V.est;
Tr.est.a=Tr.est.b=Tr.est;
tri.band=boot.image(Ya=Ya,Yb=Yb,Z=Z,d.est=d.est,d.band=d.band,r=r,V.est.a=V.est.a,Tr.est.a=Tr.est.a,V.est.b
tri.band$tri.band
V.band.a=tri.band$V.band.a; Tr.band.a=tri.band$Tr.band.a;
V.band.b=tri.band$V.band.b; Tr.band.b=tri.band$Tr.band.b;
```

---

boot1g.image *Select triangulation for one sample SCC construction using bootstrap*

Description

Select triangulation for one sample SCC construction using bootstrap

Usage

```
boot1g.image(Y, Z, d.band, r, B.est, Q2.est, K.est, B.bands, Q2.bands,
  K.bands, ind.inside, lambda, nboot = 50, alpha0, adjust.sigma = TRUE)
```

---

boot2g.image *Select triangulation for two sample SCC construction*

Description

Select triangulation for two sample SCC construction

Usage

```
boot2g.image(Ya, Yb, Z, d.band, r, B.est.a, Q2.est.a, K.est.a, B.est.b,
  Q2.est.b, K.est.b, tri.boot = NULL, B.bands, Q2.bands, K.bands,
  ind.inside, lambda, nboot = 50, alpha0 = 0.05, adjust.sigma = TRUE)
```

---

cv.image *Choose triangulation for estimating mean functions via leave-image-out cross-validation*

Description

The function selects triangulation for estimating mean function based on leave-image-out cross-validation.

Usage

```
cv.image(Y, Z, d.est = 5, r = 1, V.ests, Tr.ests, lambda, nfold = 10)
```

Arguments

`Y`	a matrix of imaging data, each row corresponding to one subject/image.
`Z`	a 2-column matrix specifying locations of each pixel/voxel.
`d.est`	degree of bivariate spline, default is 5.
`r`	smoothness parameter. Default is 1.
`V.ests`	lists of matrices containing vertices' information of triangulation candidates.
`Tr.ests`	list of 3-column matrices specifying triangles in the triangulation candidates.
`lambda`	the vector of the candidates of penalty parameter.
`nfold`	number of folds in k-fold cross-validation. Default is 10.

Details

This R package is the implementation program for manuscript entitled "Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data" by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

Examples

```
# Triangulation information;
data(Brain.V1); data(Brain.Tr1); # triangulation No. 1;
data(Brain.V2); data(Brain.Tr2); # triangulation No. 2;
data(Brain.V3); data(Brain.Tr3); # triangulation No. 3;
V.ests=list(V1=Brain.V1,V2=Brain.V2,V3=Brain.V3);
Tr.ests=list(Tr1=Brain.Tr1,Tr2=Brain.Tr2,Tr3=Brain.Tr3);
# Location information;
n1=40; n2=40;
npix=n1*n2
u1=seq(0,1,length.out=n1)
v1=seq(0,1,length.out=n2)
uu=rep(u1,each=n2)
vv=rep(v1,times=n1)
Z=as.matrix(cbind(uu,vv))
ind.inside=inVT(Brain.V1,Brain.Tr1,Z[,1],Z[,2])$ind.inside
# Parameters for bivariate spline over triangulation;
d.est=5; r=1;
# simulation parameters
n=50; lam1=0.5; lam2=0.2; mu.func=2; noise.type='Func';
lambda=10^{seq(-6,3,0.5)}
dat=data1g.image(n,Z,ind.inside,mu.func,noise.type,lam1,lam2)
Y=dat$Y
tri.est=cv.image(Y,Z,d.est,r,V.ests,Tr.ests,lambda)
tri.est$tri.select; V.est=tri.est$V.est; Tr.est=tri.est$Tr.est;
```

`data1g.image` *Generate simulation examples: one sample case*

Description

The function generates one-sample simulation data using four different mean functions and scheme described in the paper.

Usage

```
data1g.image(n, Z, ind.inside, mu.func = c(1, 2, 3, 4),
  noise.type = c("Func", "Const"), lam1, lam2, iter = 2019)
```

Arguments

n	number of images in the simulated data.
Z	a 2-column matrix specifying locations of information.
ind.inside	a vector of indices specifying which locations in matrix Z fall inside the irregular domain.

`mu.func`	a integer in `c(1,2,3,4)` that specifies what mean function is used for data generation. Value `1,2,3,4` correspond to quadratic, exponential, cubic and sine functions described in the paper respectively.
`noise.type`	'Func'/'Const' indicating heterogeneous or homogeneous variance for measurement error.
`lam1, lam2`	the eigenvalues used to adjust subject-level variation in simulation data.
`iter`	random seed.

Details

This R package is the implementation program for manuscript entitled "Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data" by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

`data2g.image` *Generate simulation examples: two sample case*

Description

The function generates two-sample simulation data using four different mean functions and eigenfunctions described in the paper.

Usage

```
data2g.image(na, nb, Z, ind.inside, mu1.func, noise.type = c("Func",
  "Const"), lam1, lam2, delta, iter = 2019)
```

Arguments

`na`	number of images/subjects in the first sample.
`nb`	number of images/subjects in the second sample.
`Z`	a 2-column matrix specifying locations of information.
`ind.inside`	a vector of indices specifying which locations in matrix Z fall inside the irregular domain.
`mu1.func`	a integer in `c(1,2,3,4)` that specifies what mean function is used for data generation of the first sample. Value `1,2,3,4` correspond to quadratic, exponential, cubic and sine functions described in the paper respectively.
`noise.type`	'Func'/'Const' indicating heterogeneous or homogeneous variance for measurement error.
`lam1, lam2`	the eigenvalues used to adjust subject-level variation in simulation data.
`delta`	a parameter indicating the scale of difference between two samples' mean functions.
`iter`	random seed.

Details

This R package is the implementation program for manuscript entitled "Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data" by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

`fit.image` *Estimate the mean function via bivariate penalized spline over triangulation*

Description

This function is used to fit the mean function of the imaging data via bivariate penalized spline over triangulation. The tuning parameter is selected by generalized cross validation.

Usage

```
fit.image(Y, Z, V.est, Tr.est, d.est = 5, r = 1, lambda = 10^(-6:3),
  proj.matrix = FALSE)
```

Arguments

`Y`	a matrix of imaging data, each row corresponding to one subject/image.
`Z`	a 2-column matrix specifying locations of each pixel/voxel.
`V.est`	the 2-column matrix of vertices' coordinates in the triangulation for estimating mean function of the first set of imaging data.
`Tr.est`	the 3-column matrix of indices of the vertices of triangles in the triangulation.
`d.est`	degree of bivariate spline, default is 5.
`r`	smoothness parameter. Default is 1.
`lambda`	candidate of the penalty parameter. Default is `10^(-6:3)`.
`proj.matrix`	a logical value indicating whether the projection matrix will be returned for adjusting $\sigma(z)$ in the construction of SCC, default is FALSE.

Details

This R package is the implementation program for manuscript entitled "Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data" by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

Examples

```
# Triangulation information;
data(Brain.V2); data(Brain.Tr2); # triangulation No. 2;
V.est=Brain.V2; Tr.est=Brain.Tr2;
# Location information;
n1=40; n2=40;
npix=n1*n2
u1=seq(0,1,length.out=n1)
v1=seq(0,1,length.out=n2)
uu=rep(u1,each=n2)
vv=rep(v1,times=n1)
Z=as.matrix(cbind(uu,vv))
ind.inside=inVT(V.est,Tr.est,Z[,1],Z[,2])$ind.inside
# Parameters for bivariate spline over triangulation;
d.est=5; r=1;
# simulation parameters
```

```
n=50; lam1=0.5; lam2=0.2; mu.func=2; noise.type='Func';
lambda=10^{seq(-6,3,0.5)}
dat=data1g.image(n,Z,ind.inside,mu.func,noise.type,lam1,lam2)
Y=dat$Y
Ym=matrix(apply(Y,2,mean),nrow=1)
mfit=fit.image(Ym,Z,V.est,Tr.est,d.est,r,lambda)
plot(mfit)
```

---

fit.mean *Penalized least square estimation with GCV*

Description

The function selects penalty parameter via generalized cross validation and solves penalized least square problem in estimating mean functions with bivariate penalized splines.

Usage

```
fit.mean(B, Q2, K, lambda = 10^(-6:3), Y, proj.matrix = FALSE)
```

Arguments

B	bivariate spline basis matrix.
Q2	qr decomposition of the smoothness matrix.
K	energy matrix.
lambda	candidate of the penalty parameter.
Y	a matrix of data with each row corresponding to one subject/image.
proj.matrix	a logical value indicating whether the projection matrix will be returned for adjusting $\sigma(z)$ in the construction of SCC.

---

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

plot.image — *Plot image with triangulation*

Description

The plot method for class 'image'. The function plots a series of images of triangulations used for mean estimation, estimated mean functions and simultaneous confidence corridors.

Usage

```
## S3 method for class 'image'
plot(mfit)
```

Arguments

`mfit`	an "image" object returned from either `fit.image` or `scc.image`.

Details

This R package is the implementation program for manuscript entitled "Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data" by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

scc.image — *Simultaneous confidence corridors for mean function of imaging data*

Description

The function is used to construct SCC for mean function of one group of images or difference between mean functions of two sets of images.

Usage

```
scc.image(Ya, Yb = NULL, Z, Z.band = NULL, d.est = 5, d.band = 2,
  r, V.est.a, Tr.est.a, V.band.a, Tr.band.a, V.est.b = NULL,
  Tr.est.b = NULL, V.band.b = NULL, Tr.band.b = NULL,
  penalty = TRUE, lambda, alpha.grid = c(0.1, 0.05, 0.01),
  adjust.sigma = TRUE)
```

Arguments

`Ya`	a matrix of imaging data, each row corresponding to one subject/image.
`Yb`	an optional matrix containing the second group of imaging data. Default is `NULL`. When `Yb` is `NULL`, a one-group SCC is constructed for the mean function of `Ya`, otherwise, a two-group SCC is constructed for the difference between the mean functions of `Yb` and `Ya`.
`Z`	a 2-column matrix specifying locations of each pixel/voxel.

`Z.band`	an optional matrix specifying locations for constructing SCC. Default is `NULL`. When `Z.band` is `NULL`, the SCC is evaluated on sample locations provided by matrix `Z`.
`d.est`	degree of bivariate spline for estimating mean function, default is 5.
`d.band`	degree of bivariate spline for constructing SCC, default is 2.
`r`	smoothness parameter, default is 1.
`V.est.a`	the 2-column matrix of vertices' coordinates in the triangulation for estimating mean function of the first set of imaging data.
`Tr.est.a`	the 3-column matrix of indices of the vertices of triangles in the triangulation.
`V.band.a, Tr.band.a`	information of triangultaion for constructing SCC of first set of imaging data.
`V.est.b, Tr.est.b`	optional information of triangulation used for estimating mean function of the second sample.
`V.band.b, Tr.band.b`	optional information of triangulation for constructing SCC of second set of imaging data.
`penalty`	logical value indicating whether bivariate penalize spline should be implemented. Default is TRUE.
`lambda`	the vector of the candidates of penalty parameter.
`alpha.grid`	vector of confidence levels. Default is `c(0.1,0.05,0.01)`.
`adjust.sigma`	a logical value indicating whether $\sigma(z)$ is adjusted when constructing SCC. Default is TRUE.

Details

This R package is the implementation program for manuscript entitled "Simultaneous Confidence Corridors for Mean Functions in Functional Data Analysis of Imaging Data" by Yueying Wang, Guannan Wang, Li Wang and R. Todd Ogden.

Examples

```
# Triangulation information;
data(Brain.V1); data(Brain.Tr1); # triangulation No. 1;
data(Brain.V2); data(Brain.Tr2); # triangulation No. 2;
V.est=Brain.V2; Tr.est=Brain.Tr2;
V.band=Brain.V1; Tr.band=Brain.Tr1;
# Location information;
n1=40; n2=40;
npix=n1*n2
u1=seq(0,1,length.out=n1)
v1=seq(0,1,length.out=n2)
uu=rep(u1,each=n2)
vv=rep(v1,times=n1)
Z=as.matrix(cbind(uu,vv))
ind.inside=inVT(V.est,Tr.est,Z[,1],Z[,2])$ind.inside
# Parameters for bivariate spline over triangulation;
d.est=5; d.band=2; r=1;

# Example 1. One-group SCC;
# simulation parameters
```

```
n=50; lam1=0.5; lam2=0.2; mu.func=2; noise.type='Func';
lambda=10^{seq(-6,3,0.5)}; alpha.grid=c(0.1,0.05,0.01);
dat=data1g.image(n,Z,ind.inside,mu.func,noise.type,lam1,lam2)
Y=dat$Y; beta.true=dat$beta.true;
out1=scc.image(Ya=Y,Z=Z,V.est.a=V.est,Tr.est.a=Tr.est,V.band.a=V.band,Tr.band.a=Tr.band,d.est=d.est,d.band=
scc=out1$scc
sum((scc[,1,2]<beta.true[ind.inside]) & (scc[,2,2]>beta.true[ind.inside]))/length(ind.inside)
plot(out1)

# Example 2. Two-group SCC;
# simulation parameters
na=50; nb=60; lam1=0.5; lam2=0.2; mu1.func=1; delta=0.3;
noise.type='Func'; lambda=10^{seq(-6,3,0.5)}; alpha.grid=c(0.10,0.05,0.01);
dat=data2g.image(na,nb,Z,ind.inside,mu1.func,noise.type,lam1,lam2,delta)
Ya=dat$Ya; Yb=dat$Yb; beta.true=dat$beta.true;
beta.diff=beta.true[,2]-beta.true[,1]
V.est.a=V.est.b=V.est;
Tr.est.a=Tr.est.b=Tr.est;
V.band.a=V.band.b=V.band;
Tr.band.a=Tr.band.b=Tr.band;
out2=scc.image(Ya=Ya,Yb=Yb,Z=Z,V.est.a=V.est.a,Tr.est.a=Tr.est.a,V.band.a=V.band.a,Tr.band.a=Tr.band.a,V.es
scc=out2$scc
sum((scc[,1,2]<beta.diff[ind.inside]) & (scc[,2,2]>beta.diff[ind.inside]))/length(ind.inside)
plot(out2)
```

`scc1g.image` *Simultaneous confidence corridors for mean function of imaging data: one sample case*

Description

Simultaneous confidence corridors for mean function of imaging data: one sample case

Usage

```
scc1g.image(Y, Z, Z.band, d.est, d.band, r, B.est, Q2.est, K.est, B.band,
  Q2.band, K.band, ind.inside, B.cover1, B.cover2, ind.inside.cover,
  penalty = FALSE, lambda, alpha.grid = c(0.1, 0.05, 0.01),
  adjust.sigma = TRUE)
```

`scc2g.image` *Simultaneous confidence corridors for mean function of imaging data: two sample case*

Description

Simultaneous confidence corridors for mean function of imaging data: two sample case

Usage

```
scc2g.image(Ya, Yb, Z, Z.band, d.est, d.band, r, B.est.a, Q2.est.a,
  K.est.a, B.est.b, Q2.est.b, K.est.b, B.band.a, Q2.band.a, K.band.a,
  B.band.b, Q2.band.b, K.band.b, ind.inside, B.cover1.a, B.cover2.a,
  B.cover1.b, B.cover2.b, ind.inside.cover, penalty = TRUE, lambda,
  alpha.grid = c(0.1, 0.05, 0.01), adjust.sigma = TRUE)
```

Index