

Package ‘ssfcov2’

April 19, 2013

Type Package

Title SS-ANOVA covariance function estimation and eigenfunction estimation

Version 1.3

Date 2012-10-22

Author Daniel Fortin

Maintainer Daniel Fortin <dfortin@iastate.edu>

Description SS-ANOVA covariance function estimation and eigenfunction estimation

License MIT

Depends fields, gss, plyr, ggplot2, lattice

Collate 'R00.R' 'R01.R' 'R1.R' 'cov.fn.R' 'covf_cy.R' 'create-
g.R' 'denman.beavers.R' 'estimate_cov_function.R' 'estimate_eigenfunctions.R' 'k1.R' 'k2.R' 'k4.R' 'my.samp.cov2d.
internal.R' 'tprk.R' 'unpenalized.terms.R'

R topics documented:

cy	2
estimate_cov_function	2
estimate_eigenfunctions	3
my.samp.cov2d	3
plot_covfit	4
sfdat	5
Index	6

cy	<i>Example of a functional data set</i>
----	---

Description

Example of a functional data set

Usage

```
data(cy)
```

Format

A data frame with 250 observations on the following 3 variables.

ID a numeric vector

Time a numeric vector

X a numeric vector

estimate_cov_function	<i>Nonparametric estimate of the covariance function</i>
-----------------------	--

Description

Nonparametric estimate of the covariance function

Usage

```
estimate_cov_function(dat, marginal.knots = NULL,
  n.marginal.knots = NULL)
```

Arguments

dat	data frame which must have columns ID, Time, X. The ID column is an integer indicator variable identifying a specific curve, Time contains the points on the "time" axis where the curve is observed, and X contains the observed values.
marginal.knots	vector of locations which will be used to create a knots locations for the covariance function. See details for how knot locations for the product space are constructed. If this is specified, then n.marginal.knots should not be specified.
n.marginal.knots	integer specifying the number of locations which will be used to create a knots locations for the covariance function. See details for how knot locations for the product space are constructed. If this is specified, then marginal.knots should not be specified.

Examples

```
data(sfdat)
cov.fit <- estimate_cov_function(sfdat, n.marginal.knots=5)
```

```
estimate_eigenfunctions
```

estimates eigenfunctions using fitted covariance function

Description

estimates eigenfunctions using fitted covariance function

Usage

```
estimate_eigenfunctions(cov.fit)
```

Arguments

`cov.fit` fitted covariance function. The structure should match the output of `estimate_cov_function`

Value

list of the eigenfunctions and a vector of the corresponding eigenvalues.

Examples

```
data(sfdat)
cov.fit <- estimate_cov_function(sfdat, n.marginal.knots=5)
eig.fit <- estimate_eigenfunctions(cov.fit)
eig.fns <- eig.fit$fns
ef1 <- eig.fns[[1]]
curve(ef1)
```

```
my.samp.cov2d
```

compute sample covariance for functional data

Description

compute sample covariance for functional data

Usage

```
my.samp.cov2d(time, x, subject, wt, marginal.knots,
  n.marginal.knots, centered = TRUE, noDiag = TRUE)
```

Arguments

<code>time</code>	vector of points on the 'time' axis
<code>x</code>	vector of functional observations
<code>subject</code>	vector of integer IDs for each curve
<code>wt</code>	vector of weights. These weights are not used to calculate anything in this function. The <code>wt</code> argument here is necessary for 'bookkeeping' purposes for covariance function estimation methods that utilize these weights.

`marginal.knots` vector of knot locations on the 'time' (i.e. marginal) axis.
`n.marginal.knots` integer specifying the number of knot locations to use on the marginal domain
`centered` logical. If FALSE a smoothing spline fit will be computed to estimate the mean functions, and this mean function will be subtracted from each curve.
`noDiag` logical. If TRUE the diagonal elements of the estimated covariance matrix will be removed.

Details

These weights argument `wt` is not used to calculate anything in this function. The `wt` argument here is necessary for 'bookkeeping' purposes for covariance function estimation methods that utilize these weights.

Value

A data frame with the sample covariances. The knot locations are included at the end of the data frame.

<code>plot_covfit</code>	<i>Plot fitted covariance function</i>
--------------------------	--

Description

Plot fitted covariance function

Usage

```
plot_covfit(fit, grid.resolution = 40, onlyPen = FALSE,
  onlyUnpen = FALSE, image = FALSE, ...)
```

Arguments

`fit` fitted covariance object
`grid.resolution` integer specifying the resolution grid points where the function will be evaluated
`onlyPen` if TRUE will plot only the portion of the covariance function estimated from penalized functions
`onlyUnpen` if TRUE will plot only the portion of the covariance function estimated from unpenalized functions
`image` if TRUE will produce an image plot. By default the plot is a wireframe plot ... additional arguments sent to plotting functions. See `image.plot` and `wireframe` for details on additional arguments

Examples

```
data(sfdat)
covfit <- estimate_cov_function(sfdat, n.marginal.knots=10)
plot_covfit(covfit, image=TRUE)
```

sfdat*This is simulated functional data set*

Description

This data set contains observations from 50 random curves, each evaluated at 10 random locations with noise. This data set was simulated using the package `sfdasim`.

Usage

```
data(sfdat)
```

Format

A data frame with 500 observations on the following 5 variables.

ID a numeric vector identifying observations from the same curve

locs.Var1 a numeric vector

locs.Var2 a numeric vector

Time a numeric vector containing points on the 'time' axis where curves are evaluated

X a numeric vector of observed values

Index

*Topic **datasets**

cy, [2](#)

sfdat, [5](#)

cy, [2](#)

estimate_cov_function, [2](#)

estimate_eigenfunctions, [3](#)

my.samp.cov2d, [3](#)

plot_covfit, [4](#)

sfdat, [5](#)