Package 'fdasrvf'

January 27, 2015

Maintainer J. Derek Tucker <dtucker@stat.fsu.edu>

Author J. Derek Tucker <dtucker@stat.fsu.edu>

Version 1.4.2

License GPL-3

Title Elastic Functional Data Analysis

Description A library for functional data analysis using the square root velocity framework

Depends R (>= 2.11.0), foreach, mvtnorm, matrixcalc, numDeriv

Suggests doParallel

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-01-29 16:24:06

R topics documented:

align_fPCA . . . 2
elastic.distance . . . 3
fdasrvf . . . 4
f_to_srvf . . . 4
gauss_model . . . 5
gradient . . . 6
horizFPCA . . . 6
optimum.reparam . . . 7
outlier.detection . . . 8
rgam . . . 9
simu_data . . . 9
simu_warp . . . 10
smooth.data . . . 10
SqrtMean . . . 11
time_warping . . . 12
toy_data . . . 13
toy_warp . . . 13
vertFPCA . . . 14

Index **15**

`align_fPCA` *Group-wise function alignment and PCA Extractions*

Description

This function aligns a collection of functions while extracting pincipal components.

Usage

```
align_fPCA(f, time, num_comp = 3, showplot = T, smooth_data = FALSE,
  sparam = 25, parallel = FALSE, cores = 8)
```

Arguments

`f`	matrix (N x M) of M functions with N samples
`time`	vector of size N describing the sample points
`num_comp`	number of principal components to extract (default = 3)
`showplot`	shows plots of functions (default = T)
`smooth_data`	smooth data using box filter (default = F)
`sparam`	number of times to apply box filter (default = 25)
`parallel`	enable parallel mode using `foreach` and `doParallel` pacakge
`cores`	set number of cores to use with `doParallel` (default = 2)

Value

Returns a list containing

`f0`	original functions
`fn`	aligned functions - matrix (N x M) of M functions with N samples
`qn`	aligned srvfs - similar structure to fn
`q0`	original srvfs - similar structure to fn
`mqn`	srvf mean - vecotr of length N
`gam`	warping functions - vecotr of length N
`Dx`	cost function
`vfpca`	list containing
`q_pca`	srvf principal directions
`f_pca`	f principal directions
`latent`	latent values
`coef`	coefficients
`U`	eigenvectors

References

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_data")
out = align_fPCA(simu_data$f,simu_data$time)
```

`elastic.distance` *Calculates two elastic distance*

Description

This functions calculates the distances between functions, D_y and D_x, where function 1 is aligned to function 2

Usage

```
elastic.distance(f1, f2, time, lambda = 0)
```

Arguments

`f1`	sample function 1
`f2`	sample function 2
`time`	sample points of functions
`lambda`	controls amount of warping (default = 0)

Value

Returns a list containing

`Dy`	amplitude distance
`Dx`	phase distance

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_data")
distances = elastic.distance(simu_data$f[,1],simu_data$f[,2],simu_data$time)
```

fdasrvf *Elastic Functional Data Analysis*

Description

A library for functional data analysis using the square root velocity framework which performs pair-wise and group-wise alignment as well as modeling using functional component analysis

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

f_to_srvf *Convert to SRVF*

Description

This function converts functions to srvf

Usage

```
f_to_srvf(f, time)
```

Arguments

f	matrix of functions
time	time

Value

q matrix of srvfs

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_data")
q = f_to_srvf(simu_data$f,simu_data$time)
```

gauss_model *Gaussian model of functional data*

Description

This function models the functional data using a Gaussian model extracted from the principal components of the srvfs

Usage

```
gauss_model(fn, time, qn, gam, n = 1, sort_samples = FALSE)
```

Arguments

fn	matrix (N x M) of M aligned functions with N samples
time	vector of size N describing the sample points
qn	matrix (N x M) of M aligned srvfs
gam	warping functions
n	number of random samples (n = 1)
sort_samples	sort samples (default = F)

Value

Returns a list containing

fs	random aligned samples
gams	random warping function samples
ft	random function samples

References

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_data")
data("simu_warp")
out1 = gauss_model(simu_warp$fn,simu_data$time,simu_warp$qn,simu_warp$gam,n = 10)
```

gradient *Gradient using finite diferences*

Description

This function takes the gradient of f using finite differences

Usage

```
gradient(f, binsize)
```

Arguments

f	vector with N samples
binsize	scalar of time samples

Value

g vecotr with N samples which is the gradient of f

Examples

```
data("simu_data")
out = gradient(simu_data$f[,1],mean(diff(simu_data$time)))
```

horizFPCA *Horizontal Functional Principal Component Analysis*

Description

This function calculates vertical functional principal component analysis on aligned data

Usage

```
horizFPCA(gam, no, showplot = TRUE)
```

Arguments

gam	matrix (N x N) of M of warping functions with N time samples
no	number of prinicpal components to extract
showplot	show plots of prinipal directions (default = T)

Value

Returns a list containing

gam_pca	warping functions principal directions
psi_pca	srvf principal directions
latent	latent values
U	eigenvectors
vec	shooting vectors
mu	Karcher Mean

References

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_warp")
hfpca = horizFPCA(simu_warp$gam,no = 3)
```

---

optimum.reparam *Align two functions*

Description

This function aligns two srvf functions using Dynamic Programming

Usage

```
optimum.reparam(Q1, T1, Q2, T2, lambda = 0)
```

Arguments

Q1	srvf of function 1
T1	sample points of function 1
Q2	srvf of function 2
T2	sample points of function 2
lambda	controls amount of warping (default = 0)

Value

gam warping function

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_data")
q = f_to_srvf(simu_data$f,simu_data$time)
gam = optimum.reparam(q[,1],simu_data$time,q[,2],simu_data$time)
```

`outlier.detection` *Outlier Detection*

Description

This function calculates outlier's using geodesic distances of the SRVFs from the median

Usage

```
outlier.detection(q, time, mq, k = 1.5)
```

Arguments

`q`	matrix (N x M) of M SRVF functions with N samples
`time`	vector of size N describing the sample points
`mq`	median calcuated using `time_warping`
`k`	cutoff threshold (default = 1.5)

Value

`q_outlier`	outlier functions

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("toy_data")
data("toy_warp")
q_outlier = outlier.detection(toy_warp$q0,toy_data$time,toy_warp$mqn,k=.1)
```

rgam *Random Warping*

Description

Generates random warping functions

Usage

```
rgam(N, sigma, num)
```

Arguments

N	length of warping function
sigma	variance of warping functions
num	number of warping functions

Value

gam warping functions

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
gam = rgam(N=101, sigma=.01, num=35)
```

simu_data *Simulated two Gaussian Dataset*

Description

A functional dataset where the individual functions are given by: $y_i(t) = z_{i,1}e^{-(t-1.5)^2/2} + z_{i,2}e^{-(t+1.5)^2/2}$, $t \in [-3,3]$, $\tilde{}i = 1,2,\ldots,21$, where $z_{i,1}$ and $z_{i,2}$ are *i.i.d.* normal with mean one and standard deviation 0.25. Each of these functions is then warped according to: $\gamma_i(t) = 6(\frac{e^{a_i(t+3)/6}-1}{e^{a_i}-1}) - 3$ if $a_i \neq 0$, otherwise $\gamma_i = \gamma_{id}$ ($gamma_{id}(t) = t$) is the identity warping). The variables are as follows: f containing the 21 functions of 101 samples and time which describes the sampling

Usage

```
data("simu_data")
```

Format

A list which contains f and time

---

simu_warp *Aligned Simulated two Gaussian Dataset*

Description

A functional dataset where the individual functions are given by: $y_i(t) = z_{i,1}e^{-(t-1.5)^2/2} + z_{i,2}e^{-(t+1.5)^2/2}$, $t \in [-3,3]$, ~$i = 1,2,\ldots,21$, where $z_{i,1}$ and $z_{i,2}$ are *i.i.d.* normal with mean one and standard deviation 0.25. Each of these functions is then warped according to: $\gamma_i(t) = 6(\frac{e^{a_i(t+3)/6}-1}{e^{a_i}-1}) - 3$ if $a_i \neq 0$, otherwise $\gamma_i = \gamma_{id}$ ($gamma_{id}(t) = t$) is the identity warping). The variables are as follows: f containing the 21 functions of 101 samples and time which describes the sampling which has been aligned

Usage

```
data("simu_warp")
```

Format

A list which contains the outputs of the time_warping function

---

smooth.data *Smooth Functions*

Description

This function smooths functions using standard box filter

Usage

```
smooth.data(f, sparam)
```

Arguments

f	matrix (N x M) of M functions with N samples
sparam	number of times to run box filter

Value

fo smoothed functions

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_data")
fo = smooth.data(simu_data$f,25)
```

`SqrtMean` *SRVF transform of warping functions*

Description

This function calculates the srvf of warping functions with corresponding shooting vectors

Usage

```
SqrtMean(gam)
```

Arguments

`gam`	matrix (N x M) of M warping functions with N samples

Value

Returns a list containing

`mu`	Karcher mean psi function
`gam_mu`	Karcher mean warping function
`psi`	srvf of warping functions
`vec`	shooting vectors

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_warp")
out = SqrtMean(simu_warp$gam)
```

`time_warping` *Group-wise function alignment*

Description

This function aligns a collection of functions using the elastic square-root velocity (srvf) framework.

Usage

```
time_warping(f, time, lambda = 0, method = "mean", showplot = TRUE,
  smooth_data = FALSE, sparam = 25, parallel = FALSE, cores = 2)
```

Arguments

`f`	matrix (N x M) of M functions with N samples
`time`	vector of size N describing the sample points
`lambda`	controls the elasticity (default = 0)
`method`	warp and caluclate to Karcher Mean or Median (options = "mean" or "median", default = "mean")
`showplot`	shows plots of functions (default = T)
`smooth_data`	smooth data using box filter (default = F)
`sparam`	number of times to apply box filter (default = 25)
`parallel`	enable parallel mode using `foreach` and `doParallel` pacakge
`cores`	set number of cores to use with `doParallel` (default = 2)

Value

Returns a list containing

`f0`	original functions
`fn`	aligned functions - matrix (N x M) of M functions with N samples
`qn`	aligned srvfs - similar structure to fn
`q0`	original srvf - similar structure to fn
`fmean`	function mean or median - vector of length N
`mqn`	srvf mean or median - vector of length N
`gam`	warping functions - similar structure to fn
`orig.var`	Original Variance of Functions
`amp.var`	Amplitude Variance
`phase.var`	Phase Variance
`qun`	Cost Function Value

References

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J. S., May 2011. Registration of functional data using fisher-rao metric, arXiv:1103.3817v2 [math.ST].

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_data")
out = time_warping(simu_data$f,simu_data$time)
```

---

toy_data *Distrubted Gaussian Peak Dataset*

Description

A functional dataset where the individual functions are given by a gaussian peak with locations along the x-axis. The variables are as follows: f containing the 29 functions of 101 samples and time which describes the sampling

Usage

```
data("toy_data")
```

Format

A list which contains f and time

---

toy_warp *Aligned Distrubted Gaussian Peak Dataset*

Description

A functional dataset where the individual functions are given by a gaussian peak with locations along the x-axis. The variables are as follows: f containing the 29 functions of 101 samples and time which describes the sampling which as been aligned

Usage

```
data("toy_warp")
```

Format

A list which contains the outputs of the time_warping function

vertFPCA *Vertical Functional Principal Component Analysis*

Description

This function calculates vertical functional principal component analysis on aligned data

Usage

```
vertFPCA(fn, time, qn, no, showplot = TRUE)
```

Arguments

fn	matrix (N x M) of M aligned functions with N samples
time	vector of size N describing the sample points
qn	matrix (N x M) of M of aligned srvfs
no	number of prinicpal components to extract
showplot	show plots of prinipal directions (default = T)

Value

Returns a list containing

q_pca	srvf principal directions
f_pca	f principal directions
latent	latent values
coef	coefficients
U	eigenvectors

References

Tucker, J. D., Wu, W., Srivastava, A., May 2012. Generative Models for Function Data using Phase and Amplitude Separation, submitted to Computational Statistics and Data Analysis.

Tucker, J. D., Wu, W., Srivastava, A., Generative Models for Function Data using Phase and Amplitude Separation, Computational Statistics and Data Analysis (2012), 10.1016/j.csda.2012.12.001.

Examples

```
data("simu_warp")
data("simu_data")
vfpca = vertFPCA(simu_warp$fn,simu_data$time,simu_warp$qn,no = 3)
```

Index

∗Topic **alignment,**
align_fPCA, 2
elastic.distance, 3
optimum.reparam, 7
smooth.data, 10
∗Topic **alignment**
f_to_srvf, 4
gradient, 6
horizFPCA, 6
SqrtMean, 11
time_warping, 12
vertFPCA, 14
∗Topic **datasets**
simu_data, 9
simu_warp, 10
toy_data, 13
toy_warp, 13
∗Topic **detection**
outlier.detection, 8
∗Topic **diffeomorphism,**
rgam, 9
∗Topic **distances**
elastic.distance, 3
∗Topic **function**
rgam, 9
∗Topic **outlier**
outlier.detection, 8
∗Topic **pca**
align_fPCA, 2
gauss_model, 5
optimum.reparam, 7
smooth.data, 10
∗Topic **srvf**
align_fPCA, 2
elastic.distance, 3
f_to_srvf, 4
gradient, 6
horizFPCA, 6
optimum.reparam, 7
outlier.detection, 8
smooth.data, 10
SqrtMean, 11
time_warping, 12
vertFPCA, 14
∗Topic **warping**
rgam, 9

align_fPCA, 2

elastic.distance, 3

f_to_srvf, 4
fdasrvf, 4
fdasrvf-package *(*fdasrvf*)*, 4
foreach, 2, *12*

gauss_model, 5
gradient, 6

horizFPCA, 6

optimum.reparam, 7
outlier.detection, 8

rgam, 9

simu_data, 9
simu_warp, 10
smooth.data, 10
SqrtMean, 11

time_warping, *8*, 12
toy_data, 13
toy_warp, 13

vertFPCA, 14