Package ‘fdasrvf’

April 30, 2017

Type Package

Title Elastic Functional Data Analysis

Version 1.8.1

Date 2017-4-17

Author J. Derek Tucker <jdtuck@sandia.gov>

Maintainer J. Derek Tucker <jdtuck@sandia.gov>

Description Performs alignment, PCA, and modeling of multidimensional and unidimensional functions using the square-root velocity framework (Srivastava et al., 2011 <arXiv:1103.3817> and Tucker et al., 2014 <DOI:10.1016/j.csda.2012.12.001>). This framework allows for elastic analysis of functional data through phase and amplitude separation.

License GPL-3

LazyData TRUE

SystemRequirements C++11

Imports Rcpp (>= 0.12.1), coda, foreach, mvtnorm, matrixcalc, splines, parallel, fields, doParallel, viridisLite

Suggests akima, plot3D, plot3Drgl, rgl

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 3.1.0),

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-04-30 19:10:57 UTC

R topics documented:

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, MaxItr = 51)
```

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)
MaxItr	maximum number of iterations

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,MaxItr = 1)  # use more iterations for accuracy
```

AmplitudeBoxplot *Amplitude Boxplot*

Description

This function constructs the amplitude boxplot

Usage

```
AmplitudeBoxplot(fn, fmedian, qn, qmedian, time, alpha = 0.05, ka = 1,
  showplot = T)
```

Arguments

fn	matrix (N x M) of M aligned functions with N samples
fmedian	vector of M samples of the median calculated using time_warping with median
qn	matrix (N x M) of M of aligned srvfs
qmedian	vector of M samples of the median calculated using time_warping with median
time	vector of size N describing the sample points
alpha	quantile value (default=.05, i.e., 95%)
ka	scalar for outlier cutoff (default=1)
showplot	shows plots of functions (default = T)

Value

Returns a list containing

median_y	median function
Q1	First quartile
Q3	Second quartile
Q1a	First quantile based on alpha
Q3a	Second quantile based on alpha
minn	minimum extreme function
maxx	maximum extreme function
outlier_index	indexes of outlier functions

References

Xie, W., S. Kurtek, K. Bharath, and Y. Sun (2016). "A Geometric Approach to Visualization of Variability in Functional Data." Journal of the American Statistical Association in press: 1-34.

Examples

```
data("simu_warp_median")
data("simu_data")
out <- AmplitudeBoxplot(simu_warp_median$fn, simu_warp_median$fmean,
                        simu_warp_median$qn, simu_warp_median$mqn, simu_data$time)
```

---

beta *MPEG7 Curve Dataset*

Description

Contains the MPEG7 curve data set which is 20 curves in 65 classes. The array is structured with dimension (2,100,65,20)

Usage

```
data("mpeg7")
```

Format

an array of shape (2,100,65,20)

---

calc_shape_dist *Elastic Shape Distance*

Description

Calculate elastic shape distance between two curves beta1 and beta2

Usage

```
calc_shape_dist(beta1, beta2, mode = "O")
```

Arguments

beta1	array describing curve1 (n,T)
beta2	array describing curve
mode	Open ("O") or Closed ("C") curves

Value

d geodesic distance

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
d = calc_shape_dist(beta[,,1,1],beta[,,1,4])
```

`curve_geodesic` *Form geodesic between two curvese*

Description

Form geodesic between two curves using Elastic Method

Usage

```
curve_geodesic(beta1, beta2, k = 5)
```

Arguments

`beta1`	array describing curve 1 (n,T)
`beta2`	array describing curve 2 (n,T)
`k`	number of curves along geodesic (default 5)

Value

a list containing

`geod`	curves along geodesic (n,T,k)
`geod_q`	srvf's along geodesic

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
out = curve_geodesic(beta[,,1,1], beta[,,1,5])
```

curve_karcher_cov *Curve Karcher Covariance*

Description

Calculate Karcher Covariance of a set of curves

Usage

```
curve_karcher_cov(betamean, beta, mode = "O")
```

Arguments

betamean	array (n,T) of mean curve
beta	array (n,T,N) for N number of curves
mode	Open ("O") or Closed ("C") curves

Value

K covariance matrix

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
out = curve_srvf_align(beta[,,1,1:2],maxit=2) # note: use more shapes, small for speed
K = curve_karcher_cov(out$betamean, beta[,,1,1:2])
```

curve_karcher_mean *Karcher Mean of Curves*

Description

Calculates Karcher mean of a collection of curves using the elastic square-root velocity (srvf) framework.

Usage

```
curve_karcher_mean(beta, mode = "O", maxit = 20)
```

Arguments

`beta`	array (n,T,N) for N number of curves
`mode`	Open ("O") or Closed ("C") curves
`maxit`	maximum number of iterations

Value

Returns a list containing

`mu`	mean srvf
`betamean`	mean curve
`v`	shooting vectors
`q`	array of srvfs

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
out = curve_karcher_mean(beta[,,1,1:2],maxit=2) # note: use more shapes, small for speed
```

`curve_pair_align` *Pairwise align two curves*

Description

This function aligns to curves using Elastic Framework

Usage

```
curve_pair_align(beta1, beta2)
```

Arguments

`beta1`	array describing curve 1 (n,T)
`beta2`	array describing curve 2 (n,T)

Value

a list containing

beta2n	aligned curve 2 to 1
q2n	aligned srvf 2 to 1
gam	warping function
q1	srvf of curve 1

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
out = curve_pair_align(beta[,,1,1], beta[,,1,5])
```

---

curve_principal_directions

Curve PCA

Description

Calculate principal directions of a set of curves

Usage

```
curve_principal_directions(betamean, mu, K, mode = "O", no = 3, N = 5)
```

Arguments

betamean	array (n,T) of mean curve
mu	array (n,T) of mean srvf
K	array (2*T,2*T) covariance matrix
mode	Open ("O") or Closed ("C") curves
no	number of components
N	number of samples on each side of mean

Value

pd list describing principal directions

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
out = curve_srvf_align(beta[,,1,1:2],maxit=2) # note: use more shapes, small for speed
K = curve_karcher_cov(out$betamean, beta[,,1,1:2])
pd = curve_principal_directions(out$betamean, out$q_mu, K)
```

---

curve_srvf_align *Align Curves*

Description

Aligns a collection of curves using the elastic square-root velocity (srvf) framework.

Usage

```
curve_srvf_align(beta, mode = "O", maxit = 20)
```

Arguments

beta	array (n,T,N) for N number of curves
mode	Open ("O") or Closed ("C") curves
maxit	maximum number of iterations

Value

Returns a list containing

betan	aligned curves
qn	aligned srvfs
betamean	mean curve
q_mu	mean SRVFs

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
out = curve_srvf_align(beta[,,1,1:2],maxit=2) # note: use more shapes, small for speed
K = curve_karcher_cov(out$betamean, beta[,,1,1:2])
```

curve_to_q *Convert to SRVF space*

Description

This function converts curves to SRVF

Usage

```
curve_to_q(beta)
```

Arguments

beta	array describing curve (n,T)

Value

q array describing srvf

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
q = curve_to_q(beta[,,1,1])
```

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)
```

---

elastic.logistic *Elastic Logistic Regression*

Description

This function identifies a logistic regression model with phase-variablity using elastic methods

Usage

```
elastic.logistic(f, y, time, B = NULL, df = 20, max_itr = 20,
  smooth_data = FALSE, sparam = 25, parallel = FALSE, cores = 2)
```

Arguments

f	matrix (N x M) of M functions with N samples
y	vector of size M labels (1/-1)
time	vector of size N describing the sample points
B	matrix defining basis functions (default = NULL)
df	scalar controling degrees of freedom if B=NULL (default=20)
max_itr	scalar number of iterations (default=20)

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

alpha	model intercept
beta	regressor function
fn	aligned functions - matrix (N x M) of M functions with N samples
qn	aligned srvfs - similar structure to fn
gamma	warping functions - similar structure to fn
q	original srvf - similar structure to fn
B	basis matrix
b	basis coefficients
Loss	logistic loss
type	model type ('logistic')

References

Tucker, J. D., Wu, W., Srivastava, A., Elastic Functional Logistic Regression with Application to Physiological Signal Classification, Electronic Journal of Statistics (2014), submitted.

---

elastic.mlogistic *Elastic Multinomial Logistic Regression*

Description

This function identifies a multinomial logistic regression model with phase-variablity using elastic methods

Usage

```
elastic.mlogistic(f, y, time, B = NULL, df = 20, max_itr = 20,
  smooth_data = FALSE, sparam = 25, parallel = FALSE, cores = 2)
```

Arguments

`f`	matrix (N x M) of M functions with N samples
`y`	vector of size M labels 1,2,...,m for m classes
`time`	vector of size N describing the sample points
`B`	matrix defining basis functions (default = NULL)
`df`	scalar controling degrees of freedom if B=NULL (default=20)
`max_itr`	scalar number of iterations (default=20)
`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

`alpha`	model intercept
`beta`	regressor function
`fn`	aligned functions - matrix (N x M) of M functions with N samples
`qn`	aligned srvfs - similar structure to fn
`gamma`	warping functions - similar structure to fn
`q`	original srvf - similar structure to fn
`B`	basis matrix
`b`	basis coefficients
`Loss`	logistic loss
`type`	model type ('mlogistic')

References

Tucker, J. D., Wu, W., Srivastava, A., Elastic Functional Logistic Regression with Application to Physiological Signal Classification, Electronic Journal of Statistics (2014), submitted.

`elastic.prediction` *Elastic Predicition from Regression Models*

Description

This function performs prediction from an elastic regression model with phase-variablity

Usage

```
elastic.prediction(f, time, model, y = NULL, smooth_data = FALSE,
  sparam = 25)
```

Arguments

`f`	matrix (N x M) of M functions with N samples
`time`	vector of size N describing the sample points
`model`	list describing model from elastic regression methods
`y`	responses of test matrix f (default=NULL)
`smooth_data`	smooth data using box filter (default = F)
`sparam`	number of times to apply box filter (default = 25)

Value

Returns a list containing

`y_pred`	predicited values of f or probabilities depending on model
`SSE`	sum of squared errors if linear
`y_labels`	labels if logistic model
`PC`	probability of classification if logistic

References

Tucker, J. D., Wu, W., Srivastava, A., Elastic Functional Logistic Regression with Application to Physiological Signal Classification, Electronic Journal of Statistics (2014), submitted.

elastic.regression *Elastic Linear Regression*

Description

This function identifies a regression model with phase-variablity using elastic methods

Usage

```
elastic.regression(f, y, time, B = NULL, lam = 0, df = 20, max_itr = 20,
  smooth_data = FALSE, sparam = 25, parallel = FALSE, cores = 2)
```

Arguments

`f`	matrix (N x M) of M functions with N samples
`y`	vector of size M responses
`time`	vector of size N describing the sample points
`B`	matrix defining basis functions (default = NULL)
`lam`	scalar regularization parameter (default=0)
`df`	scalar controling degrees of freedom if B=NULL (default=20)
`max_itr`	scalar number of iterations (default=20)
`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

`alpha`	model intercept
`beta`	regressor function
`fn`	aligned functions - matrix (N x M) of M functions with N samples
`qn`	aligned srvfs - similar structure to fn
`gamma`	warping functions - similar structure to fn
`q`	original srvf - similar structure to fn
`B`	basis matrix
`b`	basis coefficients
`SSE`	sum of squared errors
`type`	model type ('linear')

References

Tucker, J. D., Wu, W., Srivastava, A., Elastic Functional Logistic Regression with Application to Physiological Signal Classification, Electronic Journal of Statistics (2014), submitted.

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.

J. D. Tucker, W. Wu, and A. Srivastava, "Phase-Amplitude Separation of Proteomics Data Using Extended Fisher-Rao Metric," Electronic Journal of Statistics, Vol 8, no. 2. pp 1724-1733, 2014.

J. D. Tucker, W. Wu, and A. Srivastava, "Analysis of signals under compositional noise With applications to SONAR data," IEEE Journal of Oceanic Engineering, Vol 29, no. 2. pp 318-330, Apr 2014.

Tucker, J. D. 2014, Functional Component Analysis and Regression using Elastic Methods. Ph.D. Thesis, Florida State University.

Robinson, D. T. 2012, Function Data Analysis and Partial Shape Matching in the Square Root Velocity Framework. Ph.D. Thesis, Florida State University.

Huang, W. 2014, Optimization Algorithms on Riemannian Manifolds with Applications. Ph.D. Thesis, Florida State University.

Cheng, W., Dryden, I. L., and Huang, X. (2016). Bayesian registration of functions and curves. Bayesian Analysis, 11(2), 447-475.

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Cheng, W., Dryden, I. L., and Huang, X. (2016). Bayesian registration of functions and curves. Bayesian Analysis, 11(2), 447-475.

Xie, W., S. Kurtek, K. Bharath, and Y. Sun (2016). "A Geometric Approach to Visualization of Variability in Functional Data." Journal of the American Statistical Association in press: 1-34.

function_group_warp_bayes

Bayesian Group Warping

Description

This function aligns a set of functions using Bayesian SRSF framework

Usage

```
function_group_warp_bayes(f, time, iter = 50000, powera = 1, times = 5,
  tau = ceiling(times * 0.04), gp = seq(dim(f)[2]), showplot = TRUE)
```

Arguments

f	matrix (N x M) of M functions with N samples
time	sample points of functions
iter	number of iterations (default = 150000)
powera	Dirchelet prior parameter (default 1)
times	factor of length of subsample points to look at (default = 5)
tau	standard deviation of Normal prior for increment (default ceil(times*.4))
gp	number of colors in plots (default seq(dim(f)[2]))
showplot	shows plots of functions (default = T)

Value

Returns a list containing

f0	original functions
f_q	f aligned quotient space
gam_q	warping functions quotient space
f_a	f aligned ambient space
gam_a	warping ambient space
qmn	mean srsf

References

Cheng, W., Dryden, I. L., and Huang, X. (2016). Bayesian registration of functions and curves. Bayesian Analysis, 11(2), 447-475.

Examples

```
## Not run:
data("simu_data")
out = function_group_warp_bayes(simu_data$f, simu_data$time)

## End(Not run)
```

function_mean_bayes *Bayesian Karcher Mean Calculation*

Description

This function calculates karcher mean of functions using DP fast for bayes method

Usage

```
function_mean_bayes(f, time, times = 5, group = 1:dim(f)[2],
  showplot = TRUE)
```

Arguments

f	matrix (N x M) of M functions with N samples
time	sample points of functions
times	factor of length of subsample points to look at (default = 5)
group	(default 1:dim(f)[2])
showplot	shows plots of functions (default = T)

Value

Returns a list containing

distfamily	dist matrix
match.matrix	matrix of warping functions
position	position
mu_5	function mean
rtmatrix	rtmatrix
sumdist	sumdist
qt.fitted	aligned srsf functions
estimator	estimator
estimator2	estimator2
regfuncs	registered functions

References

Cheng, W., Dryden, I. L., and Huang, X. (2016). Bayesian registration of functions and curves. Bayesian Analysis, 11(2), 447-475.

Examples

```
## Not run:
data("simu_data")
out = function_mean_bayes(simu_data$f, simu_data$time)

## End(Not run)
```

f_to_srvf *Convert to SRSF*

Description

This function converts functions to srsf

Usage

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

Arguments

f	matrix of functions
time	time

Value

q matrix of SRSFs

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 SRSF functions using Dynamic Programming

Usage

```
optimum.reparam(Q1, T1, Q2, T2, lambda = 0, method = "DP", w = 0.01,
  f1o = 0, f2o = 0)
```

Arguments

Q1	srsf of function 1
T1	sample points of function 1
Q2	srsf of function 2
T2	sample points of function 2
lambda	controls amount of warping (default = 0)
method	controls which optmization method (default="DP") options are Dynamic Programming ("DP"), Coordinate Descent ("DP2"), and Riemannian BFGS ("RBFGS")
w	controls LRBFGS (default = 0.01)
f1o	initial value of f1, vector or scalar depending on q1, defaults to zero
f2o	initial value of f2, vector or scalar depending on q1, defaults to zero

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)
```

pair_align_functions *Align two functions*

Description

This function aligns two functions using SRSF framework. It will align f2 to f1

Usage

```
pair_align_functions(f1, f2, time, lambda = 0, method = "DP", w = 0.01,
  f1o = 0, f2o = 0)
```

Arguments

`f1`	function 1
`f2`	function 2
`time`	sample points of functions
`lambda`	controls amount of warping (default = 0)
`method`	controls which optmization method (default="DP") options are Dynamic Programming ("DP"), Coordinate Descent ("DP2"), Riemannian BFGS ("RBFGS") and Simultaneous Alignment ("SIMUL")
`w`	controls LRBFGS (default = 0.01)
`f1o`	initial value of f1, vector or scalar depending on q1, defaults to zero
`f2o`	initial value of f2, vector or scalar depending on q1, defaults to zero

Value

Returns a list containing

`f2tilde`	aligned f2
`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")
out = pair_align_functions(simu_data$f[,1],simu_data$f[,2],simu_data$time)
```

`pair_align_functions_bayes` *Align two functions*

Description

This function aligns two functions using Bayesian SRSF framework. It will align f2 to f1

Usage

```
pair_align_functions_bayes(f1, f2, timet, iter = 15000, times = 5,
  tau = ceiling(times * 0.4), powera = 1, showplot = TRUE)
```

Arguments

f1	function 1
f2	function 2
timet	sample points of functions
iter	number of iterations (default = 15000)
times	factor of length of subsample points to look at (default = 5)
tau	standard deviation of Normal prior for increment (default ceil(times*.4))
powera	Dirchelet prior parameter (default 1)
showplot	shows plots of functions (default = T)

Value

Returns a list containing

f1	function 1
f2_q	registered function using quotient space
gam_q	warping function quotient space
f2_a	registered fucntion using ambient space
q2_a	warping function ambient space

References

Cheng, W., Dryden, I. L., and Huang, X. (2016). Bayesian registration of functions and curves. Bayesian Analysis, 11(2), 447-475.

Examples

```
data("simu_data")
out = pair_align_functions_bayes(simu_data$f[,1], simu_data$f[,2], simu_data$time)
```

---

pair_align_image *Pairwise align two images This function aligns to images using the q-map framework*

Description

Pairwise align two images This function aligns to images using the q-map framework

Usage

```
pair_align_image(I1, I2, M = 5, ortho = TRUE, basis_type = "t",
  resizei = TRUE, N = 64, stepsize = 1e-05, itermax = 1000)
```

Arguments

`I1`	reference image
`I2`	image to warp
`M`	number of basis elements (default=5)
`ortho`	orthonormalize basis (default=TRUE)
`basis_type`	("t","s","i","o"; default="t")
`resizei`	resize image (default=TRUE)
`N`	size of resized image (default=64)
`stepsize`	gradient stepsize (default=1e-5)
`itermax`	maximum number of iterations (default=1000)

Value

Returns a list containing

`Inew`	aligned I2
`gam`	warping function

References

Q. Xie, S. Kurtek, E. Klassen, G. E. Christensen and A. Srivastava. Metric-based pairwise and multiple image registration. IEEE European Conference on Computer Vision (ECCV), September, 2014

`PhaseBoxplot` *Phase Boxplot*

Description

This function constructs the amplitude boxplot

Usage

```
PhaseBoxplot(gam, alpha = 0.05, kp = 1, showplot = T)
```

Arguments

`gam`	matrix (N x M) of M warping functions with N samples
`alpha`	quantile value (default=.05, i.e., 95%)
`kp`	scalar for outlier cutoff (default=1)
`showplot`	shows plots of functions (default = T)

Value

Returns a list containing

median_x	median warping function
Q1	First quartile
Q3	Second quartile
Q1a	First quantile based on alpha
Q3a	Second quantile based on alpha
minn	minimum extreme function
maxx	maximum extreme function
outlier_index	indexes of outlier functions

References

Xie, W., S. Kurtek, K. Bharath, and Y. Sun (2016). "A Geometric Approach to Visualization of Variability in Functional Data." Journal of the American Statistical Association in press: 1-34.

Examples

```
data("simu_warp_median")
out <- PhaseBoxplot(simu_warp_median$gam)
```

---

q_to_curve *Convert to curve space*

Description

This function converts SRVFs to curves

Usage

```
q_to_curve(q)
```

Arguments

q	array describing SRVF (n,T)

Value

beta array describing curve

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
q = curve_to_q(beta[,,1,1])
beta1 = q_to_curve(q)
```

`reparam_curve` *Align two curves*

Description

This function aligns two SRVF functions using Dynamic Programming

Usage

```
reparam_curve(beta1, beta2, lambda = 0, method = "DP", w = 0.01,
  rotated = T, isclosed = F)
```

Arguments

`beta1`	array defining curve 1
`beta2`	array defining curve 1
`lambda`	controls amount of warping (default = 0)
`method`	controls which optmization method (default="DP") options are Dynamic Programming ("DP"), Coordinate Descent ("DP2"), Riemannian BFGS ("RBFGS")
`w`	controls LRBFGS (default = 0.01)
`rotated`	boolean if rotation is desireid
`isclosed`	boolean if curve is closed

Value

return a List containing

`gam`	warping function
`R`	rotation matrix
`tau`	seed point

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
gam = reparam_curve(beta[,,1,1],beta[,,1,5])$gam
```

reparam_image *Find optimum reparameterization between two images*

Description

Finds the optimal warping function between two images using the elastic framework

Usage

```
reparam_image(It, Im, gam, b, stepsize = 1e-05, itermax = 1000,
  lmark = FALSE)
```

Arguments

It	template image matrix
Im	test image matrix
gam	initial warping array
b	basis matrix
stepsize	gradient stepsize (default=1e-5)
itermax	maximum number of iterations (default=1000)
lmark	use landmarks (default=FALSE)

Value

Returns a list containing

gamnew	final warping
Inew	aligned image
H	energy
stepsize	final stepsize

References

Q. Xie, S. Kurtek, E. Klassen, G. E. Christensen and A. Srivastava. Metric-based pairwise and multiple image registration. IEEE European Conference on Computer Vision (ECCV), September, 2014

resamplecurve *Resample Curve*

Description

This function resamples a curve to a number of points

Usage

```
resamplecurve(x, N = 100, mode = "O")
```

Arguments

x	matrix defining curve (n,T)
N	Number of samples to re-sample curve, N usually is > T
mode	Open ("O") or Closed ("C") curves

Value

xn matrix defining resampled curve

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
xn = resamplecurve(beta[,,1,1],200)
```

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)
```

`sample_shapes` *Sample shapes from model*

Description

Sample shapes from model

Usage

```
sample_shapes(mu, K, mode = "O", no = 3, numSamp = 10)
```

Arguments

`mu`	array (n,T) of mean srvf
`K`	array (2*T,2*T) covariance matrix
`mode`	Open ("O") or Closed ("C") curves
`no`	number of principal components
`numSamp`	number of samples

Value

samples list of sample curves

References

Srivastava, A., Klassen, E., Joshi, S., Jermyn, I., (2011). Shape analysis of elastic curves in euclidean spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (7), 1415-1428.

Examples

```
data("mpeg7")
out = curve_srvf_align(beta[,,1,1:2],maxit=2) # note: use more shapes, small for speed
K = curve_karcher_cov(out$betamean, beta[,,1,1:2])
samples = sample_shapes(out$q_mu, K)
```

---

simu_data *Simulated two Gaussian Dataset*

Description

A functional dataset where the individual functions are given by: $y_i(t) = z_{i,1}e^{-(t-1.5)^2/2} + z_{i,2}e^{-(t+1.5)^2/2}$, $t \in [-3,3]$, $i = 1, 2, \ldots, 21$, 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

---

simu_warp_median *Aligned Simulated two Gaussian Dataset using Median*

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_median")
```

Format

A list which contains the outputs of the time_warping function finding the median

---

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 and finds the mean

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)
```

SqrtMedian *SRVF transform of warping functions*

Description

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

Usage

```
SqrtMedian(gam)
```

Arguments

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

Value

Returns a list containing

median	Karcher median psi function
gam_median	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_median")
out = SqrtMedian(simu_warp_median$gam)
```

`srsf_to_f` *Convert SRSF to f*

Description

This function converts SRSFs to functions

Usage

```
srsf_to_f(q, time, f0 = 0)
```

Arguments

`q`	matrix of srsf
`time`	time
`f0`	initial value of f

Value

f matrix of 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")
q = f_to_srvf(simu_data$f,simu_data$time)
f = srsf_to_f(q,simu_data$time,simu_data$f[1,])
```

`time_warping` *Group-wise function alignment*

Description

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

Usage

```
time_warping(f, time, lambda = 0, method = "mean", showplot = TRUE,
  smooth_data = FALSE, sparam = 25, parallel = FALSE, omethod = "DP",
  MaxItr = 20)
```

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 (default=F)
omethod	optimization method (DP,DP2,RBFGS)
MaxItr	maximum number of iterations

Value

Returns a list containing

f0	original functions
fn	aligned functions - matrix (N x M) of M functions with N samples
qn	aligned SRSFs - similar structure to fn
q0	original SRSF - similar structure to fn
fmean	function mean or median - vector of length N
mqn	SRSF 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")
# use more iterations for accuracy
out = time_warping(simu_data$f,simu_data$time, MaxItr=1)
```

toy_data — *Distributed 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 Distributed 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., 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)
```

`warp_f_gamma` *Warp Function*

Description

This function warps function f by γ

Usage

```
warp_f_gamma(f, time, gamma)
```

Arguments

`f`	vector function
`time`	time
`gamma`	vector warping function

Value

fnew warped 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")
fnew = warp_f_gamma(simu_data$f[,1],simu_data$time,seq(0,1,length.out=101))
```

`warp_q_gamma` *Warp SRSF*

Description

This function warps srsf q by γ

Usage

```
warp_q_gamma(q, time, gamma)
```

Arguments

q	vector
time	time
gamma	vector warping function

Value

qnew warped 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)
qnew = warp_q_gamma(q[,1],simu_data$time,seq(0,1,length.out=101))
```

Index