

# Package ‘warpMix’

February 15, 2017

**Title** Mixed Effects Modeling with Warping for Functional Data Using B-Spline

**Version** 0.1.0

**Author** Emilie Devijver, Gerda Claeskens, Irene Gijbels

**Maintainer** Emilie Devijver <emilie.devijver@kuleuven.be>

**Description** Mixed effects modeling with warping for functional data using B-spline. Warping coefficients are considered as random effects, and warping functions are general functions, parameters representing the projection onto B-spline basis of a part of the warping functions. Warped data are modelled by a linear mixed effect functional model, the noise is Gaussian and independent from the warping functions.

**License** GPL (>= 3)

**LazyData** TRUE

**Imports** fda (>= 2.4.4), fields (>= 8.4-1), MASS (>= 7.3-44), reshape2 (>= 1.4.2), nlme (>= 3.1-128), lme4 (>= 1.1-12)

**Depends** R (>= 3.3.2)

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-02-15 15:11:39

## R topics documented:

criterion . . . . .	2
estimationTheta . . . . .	2
initialisationPara . . . . .	3
initialisationTheta . . . . .	3
majPara . . . . .	4
predictionTheta . . . . .	5
warpMix . . . . .	5
warpTimeFunction . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

criteron	<i>Compute the empirical L<sub>2</sub> distance related to the warping function.</i>
----------	--

---

**Description**

This function returns the empirical L<sub>2</sub> between two functions, the first one being warped.

**Usage**

```
criteron(t, f, g, theta, splineBasisW)
```

**Arguments**

t	A vector of numbers, corresponding to time points.
f	A vector of numbers, corresponding to the evaluated function.
g	A vector of numbers, corresponding to the evaluated function.
theta	A vector of warping parameters.
splineBasisW	A matrix, corresponding to the spline basis for the warping functions, evaluated in time points.

**Value**

A list, with crit the distance.

---

estimationTheta	<i>Estimate the warping parameters.</i>
-----------------	---

---

**Description**

This function estimate the warping parameters, knowing the observations and the individual aligned curves.

**Usage**

```
estimationTheta(t, y, splineBasisW, indSignal, thetaObs)
```

**Arguments**

t	A vector of numbers, corresponding to time points.
y	A matrix of numbers, corresponding to observations (size: T * n).
splineBasisW	A matrix, corresponding to the spline basis for the warping functions, evaluated in time points.
indSignal	A matrix, corresponding to the individual aligned curves.
thetaObs	A matrix, corresponding to initial values for the warping parameters.

**Value**

A list, with theta, a matrix of estimated warping parameters, and wT, the corresponding warping functions.

---

initialisationPara     *Initialize the functional parameters (associated to the aligned curves).*

---

**Description**

This function initializes the mean curve, the individual effect  $U_i$ , related to aligned curves.

**Usage**

```
initialisationPara(t, y, splineBasisMu, splineBasisU, warpTime)
```

**Arguments**

t	A vector of numbers, corresponding to time points.
y	A matrix of numbers, corresponding to observations (size: $T * n$ ).
splineBasisMu	A matrix, corresponding to the spline basis for the global mean function, evaluated in time points.
splineBasisU	A matrix, corresponding to the spline basis for the individual curves, evaluated in time points.
warpTime	A matrix, corresponding to warping time points.

**Value**

A list, with x, aligned curves, alphaMu the coefficients of the mean curve, sigmaEpsilon the variance of the noise, sigmaU the variance of the random effects, and indSignal each individual curves.

---

initialisationTheta     *Initialize the warping parameters.*

---

**Description**

This function initializes the warping parameters

**Usage**

```
initialisationTheta(t, y, splineBasisW)
```

**Arguments**

t	A vector of numbers, corresponding to time points.
y	A matrix of numbers, corresponding to observations (size: T * n).
splineBasisW	A matrix, corresponding to the spline basis for the warping functions, evaluated in time points.

**Value**

A list, with theta, a matrix of estimated warping parameters, and wT, the corresponding warping functions.

---

majPara	<i>Update the functional parameters (associated to the aligned curves).</i>
---------	---

---

**Description**

This function updates the estimations of the mean curve, the individual effect  $U_i$ , related to aligned curves.

**Usage**

```
majPara(t, y, splineBasisMu, splineBasisU, warpTime)
```

**Arguments**

t	A vector of numbers, corresponding to time points.
y	A matrix of numbers, corresponding to observations (size: T * n).
splineBasisMu	A matrix, corresponding to the spline basis for the global mean function, evaluated in time points.
splineBasisU	A matrix, corresponding to the spline basis for the individual curves, evaluated in time points.
warpTime	A matrix, corresponding to warping time points.

**Value**

A list, with x, aligned curves, alphaMu the coefficients of the mean curve, sigmaEpsilon the variance of the noise, sigmaU the variance of the random effects, and indSignal the individual curves.

---

predictionTheta	<i>Predict the warping parameters.</i>
-----------------	--

---

### Description

This function predict the warping parameters, using the estimations of those parameters, and fitting a linear mixed effect model on them.

### Usage

```
predictionTheta(thetaObs, sigmaEpsilon)
```

### Arguments

thetaObs	A matrix (size: $n * T$ ) corresponding of the estimations of the warping parameters.
sigmaEpsilon	A number, defining the variance of the noise in the linear mixed- effect model fitted on the warping parameters.

### Value

A list, with theta, a matrix of predicted warping parameters, sigmaE the covariance of the random effects, and theta0 the mean.

---

warpMix	<i>Estimate the non linear mixed effect functional model.</i>
---------	---

---

### Description

This function returns estimates of parameters in the non linear functional mixed-effect model defined to warp data and estimate the underlying model with mixed effect.

### Usage

```
warpMix(t, y, baseMu, baseU, baseW, sigmaEpsilonTilde = 10^-3,
        threshold = 10^-3, nIte = 100)
```

### Arguments

t	A vector of numbers, corresponding to time points.
y	A matrix of numbers, corresponding to observations (size: $T * n$ ).
baseMu	A B-spline used to decompose the global mean.
baseU	A B-spline used to decompose the individual effects.
baseW	A B-spline used to decompose the warping functions.

sigmaEpsilonTilde	A number, defining the variance of the noise in the linear mixed- effect model fitted on the warping parameters.
threshold	A number, defining the threshold of convergence.
nIte	Maximum number of iterations

### Details

Notice that the warping parameters are considered as random effects.

### Value

A list, with `fonct`, functional quantities (`indCurvAlign` the individual aligned curves, warping the warping functions and `theta`, the warping parameters), `para` the estimates of parameters (`alphaMu`, `sigmaU`, `theta0`, `sigmaTheta`, `sigmaEpsHat`), `dist` the criterion computed to reach the convergence, and others other values (`successAlphaMu`, `initTheta`, `initPara`, `CPUtime`).

### Examples

```
T = seq(0.5,0.841,length.out = 9)
n = 10
t = c(qnorm(T),1)
mu = cos(2*pi*t+pi/2)
library(fda)
baseMu = create.bspline.basis(c(0,max(t)), norder = 2, breaks = seq(0,1,0.5))
splineBasisMu = eval.basis(t,baseMu)
alphaMu = Data2fd(mu,argvals = t, baseMu)$coef
muApprox = (splineBasisMu) %*% alphaMu
baseU = create.bspline.basis(c(0,max(t)),norder = 2, breaks = seq(0,1,0.5))
mU = baseU$nbasis
sigmaU = diag(0.1,mU)
library(MASS)
alphaU = t(mvrnorm(n,rep(0,mU),sigmaU))
splineBasisU = eval.basis(t,baseU)
U = splineBasisU %*% alphaU
epsilon = t(mvrnorm(n,rep(0,length(t)),0.01 * diag(1,length(t))))
X = as.vector(muApprox) + U + epsilon
baseW = create.bspline.basis(c(0,max(t)), norder = 2, breaks = c(0,0.6,1))
mW = baseW$nbasis
splineBasisW = eval.basis(t,baseW)
theta = t(mvrnorm(n,rep(0,mW),diag(0.1,mW + 1e-3 * diag(1,mW))))
wtheta = matrix(rep(0,n*length(t)),ncol = n)
for (i in c(1:n)){
  wtheta[,i] = warpTimeFunction(splineBasisW,theta[,i],t)$warpTime
}
Y = matrix(0, nrow = length(t), ncol = n)
for (i in c(1:n)){
  y = approxfun(wtheta[,i],X[,i])
  Y[,i] = y(t)
}
warpMix(t,Y,baseMu, baseU, baseW, nIte = 2)
```

---

warpTimeFunction	<i>Compute the warped time points</i>
------------------	---------------------------------------

---

**Description**

This function returns warped time points for a known warping parameter theta.

**Usage**

```
warpTimeFunction(splineBasisW, theta, t)
```

**Arguments**

splineBasisW	A matrix, corresponding to the spline basis for the warping functions, evaluated in time points.
theta	A matrix, corresponding to initial values for the warping parameters.
t	A vector of numbers, corresponding to time points.

**Value**

A vector, corresponding to the warped time points.

# Index

`criterion`, 2

`estimationTheta`, 2

`initialisationPara`, 3

`initialisationTheta`, 3

`majPara`, 4

`predictionTheta`, 5

`warpMix`, 5

`warpTimeFunction`, 7