Package 'growthfd'

July 29, 2022

Title Fitting FPCA-based growth curve model

Version 0.0.0.9000

Author Ondrej Klima [aut, cre],
Miroslav Kralik [aut]

Maintainer Ondřej Klíma <iklima@fit.vutbr.cz>

Description

This package provides a method for fiting an FPCA-based growth curve model described in the paper stated bellow. This research was funded by Technology Agency of the Czech Republic (Technologická agentura České republiky), grant number TL01000394.

Citation Kralik M., Klima O., Cuta M., Malina R., Koziel S., Polcerova L., Skultetyova A., Spanel M., Kukla L., Zemcik P. Estimating Growth in Height from Limited Longitudinal Growth Data Using Full-Curves Training Dataset:A Comparison of Two Procedures of Curve Optimization-Functional Principal Component Analysis and SITAR. Children, 2021, vol. 8, n. 10., pages 934-955.

URL https://ondrej-klima.github.io/growthfd/, https://github.com/ondrej-klima/growthfd/

License LGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.2

Imports minpack.lm,
sitar (>= 1.2.0),
fda (== 2.4.8.1),
ggplot2,
parallel,
doParallel,
foreach,
flock,
MASS,
dplyr,
tidyr,
imputeTS,
zoo,
santaR

Depends R (>= 2.10)

LazyData true

Suggests rmarkdown,
    knitr

VignetteBuilder knitr

R topics documented:

growthfd	*Fit a FPCA Growth Curve Model to a population*

Description

This function fits a model to the given measured data of a population.

Usage

```
growthfd(
  data,
  x,
  y,
  id,
  model,
  verbose = 1,
  bounds = "negative",
  filename = "",
  startFromId = NULL,
  parallel = F,
  scores.filename = "parallel.txt"
)
```

Arguments

data	Data frame containing age, height and id of individuals
x	Age at measured data points
y	Height at measured data points
id	Corresponding individual's id at measured data points
model	FPCA growth model to be fitted
verbose	Verbosity
bounds	Limitation of the interval for milestones estimation, 'negative' or 'inverse'
filename	File name for saving results after each individual
startFromId	Start the evaluation from this id
parallel	(Experimental) Parallel evaluation of the model fitting
scores.filename	
	File name for continuous saving of the scores

Value

List containing individuals id and model

Examples

```
filename <- system.file("extdata", "data.csv", package="growthfd", mustWork=TRUE)
csv <- read.csv(filename)
d <- data.frame('id'=as.factor(csv[,'id']), 'x'=csv[,'age'], 'y'=csv[,'height'])
fit<-growthfd(data=d, x=x, y=y, id=id, model=model.bgs.m)
```

---

growthfd.apv   *Compute apv of model instance*

Description

This function computes apv related to the certain instance of the model described by the given parameters.

Usage

```
growthfd.apv(model, par)
```

Arguments

model   FPCA growth model

par   Params of the model, corresponding to some individual

Value

Age of maximum growth velocity

---

growthfd.ApvRegVelocity

Register a velocity curve at population apv

Description

This function registers a curve corresponding to the supplied parameters onto the population apv.

Usage

```
growthfd.ApvRegVelocity(model, par, verbose = F)
```

Arguments

model   FPCA growth model

par   Params of the model, corresponding to the individual

Value

Velocity at apv data frame

Examples

```
filename <- system.file("extdata", "data.csv", package="growthfd", mustWork=TRUE)
csv <- read.csv(filename)
d <- data.frame('id'=as.factor(csv[,'id']), 'x'=csv[,'age'], 'y'=csv[,'height'])
m <- d$id == 'John'
fit <- growthfd.fit(model.bgs.m, age=d$x[m], height=d$y[m])
data<-growthfd.ApvRegVelocity(model.bgs.m, fit$par)
```

---

growthfd.bgs.apvs *Find apvs on growth curves*

Description

This function finds ages of maximal growth velocity on the velocity curves.

Usage

```
growthfd.bgs.apvs(age, velocity)
```

Arguments

age	Vector of ages
velocity	Matrix of velocity curves
ids	Vector of individuals' ids
limits	List of limits

Value

Vector of apv values

---

growthfd.bgs.dropoutsIds.Height
List ids of individuals to be dropped from height modeling

Description

This function returns a vector containing ids of individuals with incomplete stature measurements.

Usage

```
growthfd.bgs.dropoutsIds.Height()
```

Value

Vector of ids

growthfd.bgs.eval   *Evaluate general fda splines*

Description

This function evaluates non monotone fda splines.

Usage

```
growthfd.bgs.eval(fda, age, deriv = 0)
```

Arguments

fda	Fda object
age	Vector of ages to be evaluated
deriv	Derivative of the curve

Value

Matrix of evaluated points

growthfd.bgs.evalMonotone
                                  Evaluate monotone fda splines

Description

This function evaluates the set of monotone splines.

Usage

```
growthfd.bgs.evalMonotone(fda, age, deriv = 0)
```

Arguments

fda	Fda object
age	Vector of ages to be evaluated
deriv	Derivative of the curve

Value

Matrix of evaluated points

growthfd.bgs.gather *Gather selected columns*

Description

Selects columns with given prefix for supplied ages and gathers them into a matrix together with id
and sex data

Usage

```
growthfd.bgs.gather(data, prefix = "vysk", age = NULL)
```

Arguments

data	BGS data
prefix	Columns prefix
age	Vector containing ages (optional)

Value

Gathered data

growthfd.bgs.interpolateNAs
Estimate NA values

Description

Interpolates missing values using spline method. Interpolates data from the 'value' column and join
them as the 'valuei' column.

Usage

```
growthfd.bgs.interpolateNAs(gatheredData)
```

Arguments

gatheredData	Data in gathered form

Value

Interpolated data

---

growthfd.bgs.invertTw   *Compute inverse time-warping functions*

Description

Computes inverse for given functions.

Usage

```
growthfd.bgs.invertTw(age, tw)
```

Arguments

age	Vector of ages
tw	Fda object containing the time-warping functions

Value

Fda object containing the inverse functions

---

growthfd.bgs.measurementsAge
                                List ages of measurements

Description

This function returns a vector of ages when the measurements were performed.

Usage

```
growthfd.bgs.measurementsAge()
```

Value

Vector of ages

---

growthfd.bgs.model   *Create FPCA growth model*

Description

Creates FPCA growth model from fda objects of growth functions registered on apv and inverse time warping functions.

Usage

```
growthfd.bgs.model(amplitude, itw, nharm = 6)
```

Arguments

itw	Fda object of inverse time warping functions
nharm	Number of harmonic functions for each fpca
ampitude	Fda object of registered growth functions

Value

FPCA growth model

---

growthfd.bgs.plotAll   *Plot curves in one figure*

Description

Plots all curves from given matrix into a single figure.

Usage

```
growthfd.bgs.plotAll(age, values, xlimit = NULL, ylimit = NULL)
```

Arguments

age	Vector of ages
values	Matrix containing curves as columns
xlimit	Limits for the x axis
ylimit	Limits for the y axis

Value

GGPlot2 plot

growthfd.bgs.plotIndividuals

Plot all individual curves to pdf

Description

Plots value, velocity and acceleration curves together with apvs and measured data points into separate figure for each individual. All plots are stored into a single pdf file, one figure per page.

Usage

```
growthfd.bgs.plotIndividuals(
  age,
  ids,
  apvs,
  values,
  vel,
  acc,
  data,
  filename = "plots.pdf"
)
```

Arguments

age	Vector of ages
ids	Vector containing ids
apvs	Vector containing apv for each individual
values	Matrix with acceleration curves
vel	Matrix with velocity curves
data	Matrix with original data points
filename	File name of the output pdf

growthfd.bgs.plotIndividualsPoints

Plot individual points to pdf

Description

Plots measured data points into separate figure for each individual. All plots are stored into a single pdf file, one figure per page.

Usage

```
growthfd.bgs.plotIndividualsPoints(ids, data, filename = "plots_points.pdf")
```

Arguments

`ids`	Vector containing ids
`data`	Matrix with original data points
`filename`	File name of the output pdf
`age`	Vector of ages
`apvs`	Vector containing apv for each individual
`vel`	Matrix with velocity curves
`values`	Matrix with acceleration curves

---

growthfd.bgs.registerCurvesToApvs

Register curves to the apvs

Description

Calculates the time warping functions with respect to the supplied apvs and the growth curves for the final refinement.

Computes the time-warping functions based on landmark-based registration apv and atf points to the population means

Usage

```
growthfd.bgs.registerCurvesToApvs(fdaObject, apvs, atfs)

growthfd.bgs.registerCurvesToApvs(fdaObject, apvs, atfs)
```

Arguments

`fdaObject`	Fda object contaning the curves
`apvs`	Vector containing the respective apvs
`atfs`	Vector containing the atfs

Value

FDA object containing the time warping functions

FDA object containing the time warping functions

---

growthfd.bgs.resample *Resample the data*

Description

Resamples the data without NA values to fine grid.

Usage

```
growthfd.bgs.resample(interpolatedData)
```

Arguments

interpolatedData

     Data to be resampled.

Value

Resampled data

---

growthfd.bgs.smooth *Fit the monotone spline*

Description

This function fit the monotone splines to the data.

Usage

```
growthfd.bgs.smooth(
  resampledData,
  monotone = T,
  norder = 6,
  Lfdobj = 3,
  lambda = 0.05
)
```

Arguments

resampledData   Data to be interpolated by monotone fda splines

Value

Object with fitted splines

Examples

```
data <- read.table("D:/Growth/playground/data/bgs_07.txt", header=TRUE, sep="\t", na.strings="NA", dec=".")
#data <- data[!(data$id %in% growthfd.bgs.dropoutsIds.Height()) & data$sex == "1",]
data <- data[!(data$id %in% growthfd.bgs.dropoutsIds.Height()),]
gather <- growthfd.bgs.gather(data)
interp <- growthfd.bgs.interpolateNAs(gather)
resampled <- growthfd.bgs.resample(interp)
#smoothed <- growthfd.bgs.smooth(resampled)
smoothed <- growthfd.bgs.smooth(resampled, F)

#age <- seq(10, 18, 0.05)
#m<-growthfd.bgs.evalMonotone(smoothed$fd,age,1)
#apvs<-growthfd.bgs.apvs(age,m)
#ids <- unique(data$id)

age <- seq(10, 18, 0.05)
m<-growthfd.bgs.eval(smoothed$fd,age,1)
apvs<-growthfd.bgs.apvs(age,m)
ids <- unique(data$id)

# Fix individuals with apv lower than 10
age <- seq(7, 18, 0.05)
m<-growthfd.bgs.evalMonotone(smoothed,age,1)
apvs.na <- which(is.na(apvs))
apvs[apvs.na]<-growthfd.bgs.apvs(age,m[,apvs.na])

#age <- seq(0, 18, 0.05)
#values <- growthfd.bgs.evalMonotone(smoothed,age)
#vel <- growthfd.bgs.evalMonotone(smoothed,age,1)
#acc <- growthfd.bgs.evalMonotone(smoothed,age,2)
#growthfd.bgs.plotIndividuals(age, ids, apvs, values, vel, acc, gather)
#growthfd.bgs.plotAll(age,acc, ylimit = c(-25, 25))

age <- seq(0, 18, 0.05)
values <- growthfd.bgs.eval(smoothed$fd,age)
vel <- growthfd.bgs.eval(smoothed$fd,age,1)
acc <- growthfd.bgs.eval(smoothed$fd,age,2)
growthfd.bgs.plotIndividuals(age, ids, apvs, values, vel, acc, gather, 'plot_weight.pdf')


resampled.new <- values
dim(resampled.new) <- c(prod(dim(values)), 1)
resampled.new <- cbind(rep(ids, each=dim(values)[1]), rep(age, dim(values)[2]), resampled.new[,1])
colnames(resampled.new) <- colnames(resampled)
smoothed.new <- growthfd.bgs.smooth(resampled.new, F)

tw <- growthfd.bgs.registerCurvesToApvs(smoothed, apvs)
amp<-fda::register.newfd(smoothed.new$fd,tw)
itw <- growthfd.bgs.invertTw(age, tw)

model <- growthfd.bgs.model(amp, itw$fd)

plDf <- data.frame('phasePC1'=model$warpfpca$scores[,1], 'amplitudePC1'=model$growthfpca$scores[,1], sex=fac
ggplot2::ggplot(data = plDf, ggplot2::aes(x=phasePC1, y=amplitudePC1, colour=sex)) +
  ggplot2::geom_point() +
  ggplot2::stat_ellipse(geom = "polygon",linetype = 2,ggplot2::aes(fill = sex),alpha = 0.15) +
```

```
 ggplot2::geom_rug() +
 ggplot2::stat_smooth(method = "lm", fullrange = TRUE) +
 ggplot2::scale_colour_manual(name = "Sex", values = c('blue', 'red'), labels=c('Male', 'Female')) +
 ggplot2::scale_fill_manual(name = "Sex", values = c('blue', 'red'), labels=c('Male', 'Female'))
```

---

growthfd.digits   *Preprocess a digit data*

Description

Selects data with minimal count of measurements and convert them into a data frame.

Usage

```
growthfd.digits(data, colName, minCount = 9, ncores = 4)
```

Arguments

data	A csv table containing the measurements
colName	Name of the particular column with the length data
minCount	Minimal count of measurements
ncores	Number of cores used for parallelisation

Value

Data frame with uniformly distributed data

Examples

```
data <- read.csv2('D:/growth/digits/data_muzi_all.csv', sep=";")
res <- growthfd.digits(data=data, colName='X1_2_sur', minCount=1, ncores=4)
interp <- res[[1]]
interp$value <- interp$valuei

gather <- as.data.frame(data[,c('ind', 'age', 'X1_2_sur')])
colnames(gather) <- c('id', 'age', 'value')

resampled <- growthfd.bgs.resample(interp)
smoothed <- growthfd.bgs.smooth(resampled, F)

age <- seq(10, 18, 0.05)
m<-growthfd.bgs.eval(smoothed$fd,age,1)
apvs<-growthfd.bgs.apvs(age,m)
ids <- unique(interp$id)

age<-seq(min(interp$age), max(interp$age), 0.05)
values <- growthfd.bgs.eval(smoothed$fd,age)
vel <- growthfd.bgs.eval(smoothed$fd,age,1)
acc <- growthfd.bgs.eval(smoothed$fd,age,2)
growthfd.bgs.plotIndividuals(age, ids, apvs, values, vel, acc, gather, 'plot_X1_2_sur.pdf')
growthfd.bgs.plotIndividualsPoints(unique(res[[2]]), gather, 'plot_X1_2_sur_pts.pdf')
```

growthfd.evaluate   *Generate a Discrete Growth Curve*

Description

This function evaluates a curve function for given ages. Depending on a degree of derivation, the function produces stature, velocity or acceleration curve.

Usage

```
growthfd.evaluate(x, par, model, deriv = 0)
```

Arguments

x	Ages to be evaluated
par	Parameters of the model
model	FPCA growth model
deriv	Path to the input file

Value

Y-values of the evaluated curve

growthfd.fit   *Fit a FPCA Growth Curve Model to measurements of a single individual*

Description

This function fits a model to the given measured data of a single individual.

Usage

```
growthfd.fit(model, age, height, nprint = 1)
```

Arguments

model	FPCA growth model to be fitted
age	Age at measured data points
height	Height at at measured data points
nprint	Verbosity

Value

An optimization result object

Examples

```
age <- c(6.9, 8.2, 10, 12.1)
height <- c(114, 122, 130, 141)
fit <- growthfd.fit(model.bgs.m, age=c(6.9, 8.2, 10, 12.1), height=c(114, 122, 130, 141))
x11()
growthfd.plot(model.bgs.m, fit$par)
points(age, height)
x11()
growthfd.plot(model.bgs.m, fit$par, from=0.5, deriv = 1)
x11()
growthfd.plot(model.bgs.m, fit$par, from=0.5, deriv = 2)
```

---

growthfd.modelPars   *Standardized model scores*

Description

This function returns model parameters for the individuals used for training the model.

Usage

```
growthfd.modelPars(model)
```

Arguments

model   FPCA-based growth model

Value

Matrix containing the scores

---

growthfd.plot   *Plot a Growth Curve*

Description

This function plots a stature, velocity or acceleration curve.

Usage

```
growthfd.plot(model, par, deriv = 0, from = 0, to = 18)
```

Arguments

model	FPCA growth model
par	Parameters of the model
deriv	Path to the input file
from	The lower age limit
to	The upper age limit

growthfd.plot.ApvRegVelocity
Plot a velocity curve registered at apv

Description

This function plots a velocity curve, registered at population (model) apv in comparison with the mean curve.

Usage

```
growthfd.plot.ApvRegVelocity(data)
```

Arguments

model   Data obtained using growthfd.ApvRegVelocity

Value

Velocity at apv plot

Examples

```
filename <- system.file("extdata", "data.csv", package="growthfd", mustWork=TRUE)
csv <- read.csv(filename)
d <- data.frame('id'=as.factor(csv[,'id']), 'x'=csv[,'age'], 'y'=csv[,'height'])
m <- d$id == 'John'
fit <- growthfd.fit(model.bgs.m, age=d$x[m], height=d$y[m])
data<-growthfd.ApvRegVelocity(model.bgs.m, fit$par)
p<-growthfd.plot.ApvRegVelocity(data)
x11()
p
```

growthfd.plot.RegVelocities
Plot velocity boxplots registered on apv

Description

This function plots boxplots in time of measurements, after registration of the individual on population apv.

Usage

```
growthfd.plot.RegVelocities(populationData, individualData)
```

Arguments

populationData   Data frame for population box plots

individualData   Data frame for the individual

Value

GGPlot2 plot

Examples

```
filename <- system.file("extdata", "data.csv", package="growthfd", mustWork=TRUE)
csv <- read.csv(filename)
d <- data.frame('id'=as.factor(csv[,'id']), 'x'=csv[,'age'], 'y'=csv[,'height'])
m <- d$id == 'John'
fit <- growthfd.fit(model.bgs.m, age=d$x[m], height=d$y[m])
b <- growthfd.RegVelocities(model.bgs.m, fit$par, d$x[m])
p<- growthfd.plot.RegVelocities(b$population, b$individual)
x11()
p
```

---

growthfd.plotwarps   *Plot eigenwarps and eigenamplitudes of the model*

Description

Plots the effects of individual parameters on the curve shape.

Usage

```
growthfd.plotwarps(model, deriv = 0, ylim = NULL)
```

Arguments

model	FPCA growth model
deriv	Derivation of the growth curve
ylim	Limits the scale of y axis

---

growthfd.RegVelocities

                          Prepare data for velocity boxplots registered on apv

---

Description

This function prepares data for boxplots in time of measurements, after registration of the individual on population apv.

Usage

```
growthfd.RegVelocities(model, par, ages, rndn = 0, verbose = F)
```

Arguments

model	Model
par	Parameters of the model fitted to the measurements
ages	Ages of measurements points
rndn	Count of random curves to be evaluated

Value

Data frames for population and the individual

---

growthfd.residuals   *Compute residuals*

Description

This function computes residuals between measured stature data and data generated from the growth model.

Usage

```
growthfd.residuals(x, y, par, model)
```

Arguments

x	Vector with input ages
y	Vector with target height measurements
par	Parameters of the model
model	FPCA growth model

Value

A vector of residuals

---

growthfd.std   *Generate a Curve Function*

Description

This function generates a growth curve function based on given model and parameters, describing the growth phase and amplitude.

Usage

```
growthfd.std(par, model)
```

Arguments

par	Phase and amplitude parameters
model	FPCA growth model

Value

FDA function object

---

growthfd.warpfd   *Time warping function*

Description

This function returns the time warping function corresponding to supplied model and particular parameters.

Usage

```
growthfd.warpfd(par, model)
```

Arguments

par   Parameters of the model

model   FPCA growth model

---

growthfd.warpfdInv   *Inverse time warping function*

Description

This function returns the *inverse* time warping function corresponding to supplied model and particular parameters.

Usage

```
growthfd.warpfdInv(par, model)
```

Arguments

par   Parameters of the model

model   FPCA growth model

---

model.bgs.f   *FPCA model for girls*

Description

Model trained using 167 female individuals from Brno Growth Study (BGS).

Usage

```
model.bgs.f
```

Format

An object of class list of length 3.

---

model.bgs.m   *FPCA model for boys*

Description

Model trained using 167 male individuals from Brno Growth Study (BGS).

Usage

```
model.bgs.m
```

Format

An object of class list of length 3.

Index