Package ‘lcmm’

May 29, 2017

Type Package

Title Extended Mixed Models Using Latent Classes and Latent Processes

Version 1.7.8

Date 2017-05-29

Author Cecile Proust-Lima, Viviane Philipps, Amadou Diakite and Benoit Liquet

Maintainer Cecile Proust-Lima <cecile.proust-lima@inserm.fr>

Description Estimation of various extensions of the mixed models including latent class mixed models, joint latent latent class mixed models and mixed models for curvilinear univariate or multivariate longitudinal outcomes using a maximum likelihood estimation method.

License GPL (>= 2.0)

Depends R (>= 2.14.0), survival (>= 2.37-2)

LazyLoad yes

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-05-29 14:02:49 UTC

R topics documented:

`lcmm-package` *Estimation of extended mixed models using latent classes and latent processes.*

Description

Functions for the estimation of latent class mixed models (LCMM), joint latent class mixed models (JLCM) and mixed models for curvilinear and ordinal univariate and multivariate longitudinal outcomes (with or without latent classes of trajectory). All the models are estimated in a maximum likelihood framework using an iterative algorithm. The package also provides various post fit functions.

Details

Package:	lcmm
Type:	Package
Version:	1.7.8
Date:	2017-05-29
License:	GPL (>=2.0)
LazyLoad:	yes

The package includes for the moment the estimation of :

- latent class mixed models for Gaussian longitudinal outcomes using `hlme` function,
- latent class mixed models for other quantitative, bounded quantitative (curvilinear) and discrete longitudinal outcomes using `lcmm` function,
- latent class mixed models for multivariate (possibly curvilinear) longitudinal outcomes using `multlcmm` function,
- joint latent class mixed models for a Gaussian (or curvilinear) longitudinal outcome and a right-censored (potentially left-truncated and of multiple causes) time-to-event using `Jointlcmm` function.

Please report to the maintainer any bug or comment regarding the package for future updates.

Author(s)

Cecile Proust-Lima, Viviane Philipps, Amadou Diakite and Benoit Liquet

<cecile.proust-lima@inserm.fr>

References

Proust-Lima C, Philipps V, Liquet B (2017). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: the R package lcmm. Journal of Statistical Software, 78(2), 1-56

Lin, Turnbull, McCulloch and Slate (2002). Latent class models for joint analysis of longitudinal biomarker and event process data: application to longitudinal prostate-specific antigen readings and prostate cancer. Journal of the American Statistical Association 97, 53-65.

Muthen and Shedden (1999). Finite mixture modeling with mixture outcomes using the EM algorithm. Biometrics 55, 463-9

Proust and Jacqmin-Gadda (2005). Estimation of linear mixed models with a mixture of distribution for the random-effects. Comput Methods Programs Biomed 78:165-73

Proust, Jacqmin-Gadda, Taylor, Ganiayre, and Commenges (2006). A nonlinear model with latent process for cognitive evolution using multivariate longitudinal data. Biometrics 62, 1014-24.

Proust-Lima, Dartigues and Jacqmin-Gadda (2011). Misuse of the linear mixed model when evaluating risk factors of cognitive decline. Amer J Epidemiol 174(9), 1077-88

Proust-Lima and Taylor (2009). Development and validation of a dynamic prognostic tool for prostate cancer recurrence using repeated measures of post-treatment PSA: a joint modelling approach. Biostatistics 10, 535-49.

Proust-Lima, Sene, Taylor, Jacqmin-Gadda (2014). Joint latent class models for longitudinal and time-to-event data: a review. Statistical Methods in Medical Research 23, 74-90.

Proust-Lima, Amievan Jacqmin-Gadda (2013). Analysis of multivariate mixed longitudinal data: A flexible latent process approach. Br J Math Stat Psychol 66(3), 470-87.

Verbeke and Lesaffre (1996). A linear mixed-effects model with heterogeneity in the random-effects population. Journal of the American Statistical Association 91, 217-21

cuminc — *Predicted cumulative incidence of event according to a profile of covariates*

Description

This function computes the predicted cumulative incidence of each cause of event according to a profile of covariates from a joint latent class model. Confidence bands can be computed by a Monte-Carlo method.

Usage

```
cuminc(x, time, draws = FALSE, ndraws = 2000, ...)
```

Arguments

`x`	an object inheriting from class `Jointlcmm`
`time`	a vector of times at which the cumulative incidence is calculated
`draws`	optional boolean specifying whether a Monte Carlo approximation of the posterior distribution of the cumulative incidence is computed and the median, 2.5% and 97.5% percentiles are given. Otherwise, the predicted cumulative incidence is computed at the point estimate. By default, draws=FALSE.
`ndraws`	if draws=TRUE, ndraws specifies the number of draws that should be generated to approximate the posterior distribution of the predicted cumulative incidence. By default, ndraws=2000.
`...`	further arguments, in particular values of the covariates specified in the survival part of the joint model.

Value

An object of class `cuminc` containing as many matrices as profiles defined by the covariates values. Each of these matrices contains the event-specific cumulative incidences in each latent class at the different times specified.

Author(s)

Viviane Philipps and Cecile Proust-Lima

See Also

`Jointlcmm`,`plot.Jointlcmm`,`plot.cuminc` data_hlme *Simulated dataset for hlme function*

Description

The data were simulated from a 3-latent class linear mixed model. Repeated data for 100 subjects were simulated. The three latent classes are predicted by X2 and X3. In each latent class, Y follows a linear mixed model including intercept and time both with correlated random-effects and class-specific fixed effects. In addition, X1 and X1*time have a common impact over classes on the Y trajectory.

Usage

```
data_hlme
```

Format

A data frame with 326 observations on the following 9 variables.

`ID` subject identification number

`Y` longitudinal outcome

`Time` time of measurement

`X1` binary covariate

`X2` binary covariate

`X3` binary covariate

See Also

`hlme`, `postprob`, `summary.lcmm`, `plot.predict`

data_lcmm *Simulated dataset for lcmm and Jointlcmm functions*

Description

The data were simulated from a joint latent class mixed model with 3 classes. Repeated data of 3 longitudinal outcomes (Ydep1, Ydep2, Ydep3) and censored time of event (Tevent, Event) with delayed entry (Tentry) were simulated for a total of 300 subjects. The three latent classes were predicted by the continuous covariate X3. In each latent class, the longitudinal outcome Ydep1 followed a linear mixed model including intercept, time and squared time both with correlated random-effects and class-specific fixed effects. In addition, the binary covariate X1 and its interaction with time X1:Time had a common impact (over classes) on the Ydep1 trajectory. The longitudinal ordinal outcomes Ydep2 and Ydep3 were generated from Ydep1 using threshold models with respectively 30 and 10 thresholds. In each latent class, the time of event followed a class-specific Weibull hazard with a common proportional effect of the binary covariate X2. Both time of entry Tentry and time of censoring had a uniform distribution

Usage

```
data_lcmm
```

Format

A data frame with 1678 observations over 300 different subjects and 22 variables.

`ID` subject identification number

`Ydep1` longitudinal continuous outcome

`Ydep2` longitudinal ordinal outcome with 31 levels

`Ydep3` longitudinal ordinal outcome with 11 levels

`Tentry` delayed entry for the time-to-event

`Tevent` observed time-to-event: either censoring time or time of event

`Event` indicator that Tevent is the time of event

`Time` time of measurement

`X1` binary covariate

`X2` binary covariate

`X3` continuous covariate

`X4` categorical covariate

See Also

`Jointlcmm`, `lcmm`, `hlme`

Diffepoce — *Difference of expected prognostic cross-entropy (EPOCE) estimators and its 95% tracking interval between two joint latent class models estimated with* `Jointlcmm`

Description

This function computes the difference of 2 EPOCE estimates (CVPOL or MPOL) and its 95% tracking interval between two joint latent class models estimated using `Jointlcmm` and evaluated using epoce function. Difference in CVPOL is computed when the EPOCE was previously estimated on the same dataset as used for estimation (using an approximated cross-validation), and difference in MPOL is computed when the EPOCE was previously estimated on an external dataset.

This function does not apply for the moment with multiple causes of event (competing risks).

Usage

```
Diffepoce(epoceM1,epoceM2)
```

Arguments

`epoceM1`	a first object inheriting from class `epoce`
`epoceM2`	a second object inheriting from class `epoce`

Details

From the EPOCE estimates and the individual contributions to the prognostic observed log-likelihood obtained with `epoce` function on the same dataset from two different estimated joint latent class models, the difference of CVPOL (or MPOL) and its 95% tracking interval is computed. The 95% tracking interval is:

Delta(MPOL) +/- qnorm(0.975)*sqrt(VARIANCE) for an external dataset

Delta(CVPOL) +/- qnorm(0.975)*sqrt(VARIANCE) for the dataset used in `Jointlcmm`

where Delta(CVPOL) (or Delta(MPOL)) is the difference of CVPOL (or MPOL) of the two joint latent class models, and VARIANCE is the empirical variance of the difference of individual contributions to the prognostic observed log-likelihoods of the two joint latent class models.

See Commenges et al. (2012) and Proust-Lima et al. (2012) for further details.

Value

`call.Jointlcmm1`	the `Jointlcmm` call for epoceM1
`call.Jointlcmm2`	the `Jointlcmm` call for epoceM2
`call`	the matched call
`DiffEPOCE`	Dataframe containing, for each prediction time s, the difference in either MPOL or CVPOL depending on the dataset used, and the 95% tracking bands (TIinf and TIsup)
`new.data`	a boolean for internal use only, which is FALSE if computation is done on the same data as for `Jointlcmm` estimation, and TRUE otherwise.

Author(s)

Cecile Proust-Lima and Amadou Diakite

References

Commenges, Liquet and Proust-Lima (2012). Choice of prognostic estimators in joint models by estimating differences of expected conditional Kullback-Leibler risks. Biometrics 68(2), 380-7.

Proust-Lima, Sene, Taylor, Jacqmin-Gadda (2014). Joint latent class models for longitudinal and time-to-event data: a review. Statistical Methods in Medical Research 23, 74-90.

See Also

`Jointlcmm`,`epoce`,`summary.Diffepoce`

Examples

```
## Not run:
#### estimation with 2 latent classes (ng=2)
m2 <- Jointlcmm(fixed= Ydep1~Time*X1,random=~Time,mixture=~Time,subject='ID'
,survival = Surv(Tevent,Event)~ X1+X2 ,hazard="Weibull"
,hazardtype="PH",ng=2,data=data_lcmm,
B=c( 0.7608, -9.4974,  1.0242,  1.4331,  0.1063 , 0.6714, 10.4679, 11.3178,
 -2.5671, -0.5386,  1.4616, -0.0605,  0.9489,  0.1020,  0.2079,  1.5045),logscale=TRUE)
m1 <- Jointlcmm(fixed= Ydep1~Time*X1,random=~Time,subject='ID'
,survival = Surv(Tevent,Event)~ X1+X2 ,hazard="Weibull"
,hazardtype="PH",ng=1,data=data_lcmm,
B=c(-7.6634,  0.9136,  0.1002,  0.6641, 10.5675, -1.6589,  1.4767, -0.0806,
  0.9240,0.5643,  1.2277,  1.5004))

## EPOCE computation for predictions times from 1 to 6 on the dataset used
## for estimation of m.
VecTime <- c(1,3,5,7,9,11,13,15)
cvpol1 <- epoce(m1,var.time="Time",pred.times=VecTime)
cvpol1
cvpol2 <- epoce(m2,var.time="Time",pred.times=VecTime)
cvpol2
DeltaEPOCE <- Diffepoce(cvpol1,cvpol2)
summary(DeltaEPOCE)
plot(DeltaEPOCE,bty="l")

## End(Not run)
```

`dynpred` *Individual dynamic predictions from a joint latent class model*

Description

This function computes individual dynamic predictions and 95% confidence bands. Given a joint latent class model, a landmark time s, a horizon time t and measurements until time s, the predicted probability of event in the window [s,s+t] is calculated. Confidence bands can be provided using a Monte Carlo method.

Usage

```
dynpred(model,newdata,event=1,landmark,horizon,var.time,
 fun.time=identity,na.action=1,draws=FALSE,ndraws=2000)
```

Arguments

`model` an object inheriting from class `Jointlcmm`.

`newdata` a data frame containing the data from which predictions are computed. This data frame must contain all the model's covariates, the observations of the longitudinal and survival outcomes, the subject identifier and if necessary the variables specified in prior and TimeDepVar argumentsfrom Jointlcmm.

`event`	integer giving the event for which the prediction is to be calculated
`landmark`	a numeric vector containing the landmark times.
`horizon`	a numeric vector containing the horizon times.
`var.time`	a character indicating the time variable in `newdata`
`fun.time`	an optional function. This is only required if the time scales in the longitudinal part of the model and the survival part are different. In that case, `fun.time` is the function that translates the times from the longitudinal part into the time scale of the survival part. The default is the identity function which means that the two time scales are the same.
`na.action`	Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.
`draws`	optional boolean specifying whether median and confidence bands of the predicted values should be computed (TRUE). IF TRUE, a Monte Carlo approximation of the posterior distribution of the predicted values is computed and the median, 2.5% and 97.5% percentiles are given. Otherwise, the predicted values are computed at the point estimate. By default, draws=FALSE.
`ndraws`	if draws=TRUE, ndraws specifies the number of draws that should be generated to approximate the posterior distribution of the predicted values. By default, ndraws=2000.

Value

A list containing :

`pred`	a matrix with 4 columns if draws=FALSE and 6 columns if draws=TRUE, containing the subjects identifier, the landmark times, the horizon times, the predicted probability (if draws=FALSE) or the median, 2.5% and 97.5 % percentiles of the 'ndraws' probabilities calculated (if draws=TRUE). If a subject has no measurement before time s or if the event has already occured at time s, his probability is NA.
`newdata`	a data frame obtained from argument newdata containing time measurements and longitudinal observations used to compute the predictions

Author(s)

Cecile Proust-Lima, Viviane Philipps

References

Proust-Lima, Sene, Taylor and Jacqmin-Gadda (2014). Joint latent class models of longitudinal and time-to-event data: a review. Statistical Methods in Medical Research 23, 74-90.

See Also

`plot.dynpred`,`Jointlcmm`,`predictY`,`plot.predict`

Examples

```
## Joint latent class model with 2 classes :
m32 <- Jointlcmm(Ydep1~Time*X1,mixture=~Time,random=~Time,subject="ID",
classmb=~X3,ng=2,survival=Surv(Tevent,Event)~X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",data=data_lcmm,
B = c(0.641, -0.6217, 0, 0, 0.5045, 0.8115, -0.4316, 0.7798, 0.1027,
0.7704, -0.0479, 10.4257, 11.2972, -2.5955, -0.5234, 1.4147,
-0.05, 0.9124, 0.0501, 0.2138, 1.5027))

## Predictions at landmark 10 and 12 for horizon 3, 5 and 10 for two subjects :

dynpred(m32,landmark=c(10,12),horizon=c(3,5,10),var.time="Time",
fun.time=function(x){10*x},newdata=data_lcmm[1:8,])
## Not run:
dynpred(m32,landmark=c(10,12),horizon=c(3,5,10),var.time="Time",
fun.time=function(x){10*x},newdata=data_lcmm[1:8,],draws=TRUE,ndraws=2000)

## End(Not run)
```

`epoce` *Estimators of the Expected Prognostic Observed Cross-Entropy (EPOCE) for evaluating predictive accuracy of joint latent class models estimated using* `Jointlcmm`

Description

This function computes estimators of the Expected Prognostic Observed Cross-Entropy (EPOCE) for evaluating the predictive accuracy of joint latent class models estimated using `Jointlcmm`. On the same data as used for estimation of the `Jointlcmm` object, this function computes both the Mean Prognostic Observed Log-Likelihood (MPOL) and the Cross-Validated Observed Log-Likelihood (CVPOL), two estimators of EPOCE. The latter corrects the MPOL estimate for over-optimism by approximated cross-validation. On external data, this function only computes the Mean Prognostic Observed Log-Likelihood (MPOL).

This function does not apply for the moment with multiple causes of event (competing risks).

Usage

```
epoce(model,pred.times,var.time,fun.time=identity,newdata = NULL,subset=NULL,na.action=1)
```

Arguments

`model`	an object inheriting from class `Jointlcmm`
`pred.times`	Vector of times of prediction, from which predictive accuracy is evaluated (only subjects still at risk at the time of prediction are included in the computation, and only information before the time of prediction is considered.
`var.time`	Name of the variable indicating time in the dataset

`fun.time`	an optional function. This is only required if the time scales in the longitudinal part of the model and the survival part are different. In that case, `fun.time` is the function that translates the times from the longitudinal part into the time scale of the survival part. The default is the identity function which means that the two time scales are the same.
`newdata`	optional. When missing, the data used for estimating the `Jointlcmm` object are used, and CVPOL and MPOL are computed (internal validation). When newdata is specified, only MPOL is computed on this newdataset (external validation).
`subset`	a specification of the rows to be used: defaults to all rows. This can be any valid indexing vector for the rows of data or if that is not supplied, a data frame made up of the variable used in formula.
`na.action`	Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.

Details

EPOCE assesses the prognostic information of a joint latent class model. It relies on information theory.

MPOL computed at time s equals minus the mean individual contribution to the conditional log-likelihood of the time to event given the longitudinal data up to the time of prediction s and given the subject is still at risk of event in s.

CVPOL computed at time s equals MPOL at time s plus a penalty term that corrects for over-optimism when computing predictive accuracy measures on the same dataset as used for estimation. This penalty term is computed from the inverse of the Hessian of the joint log-likelihood and the product of the gradients of the contributions to respectively the joint log-likelihood and the conditional log-likelihood.

The theory of EPOCE and its estimators MPOL and CVPOL is given in Commenges et al. (2012), and further detailed and illustrated for joint models in Proust-Lima et al. (2013).

Value

`call.Jointlcmm`	the `Jointlcmm` call
`call.epoce`	the matched call
`EPOCE`	Dataframe containing, for each prediction time s, the number of subjects still at risk at s (and with at least one measure before s), the number of events after time s, the MPOL, and the CVPOL when computation is done on the dataset used for `Jointlcmm` estimation
`IndivContrib`	Individual contributions to the prognostic observed log-likelihood at each time of prediction. Used for computing tracking intervals of EPOCE differences between models.
`new.data`	a boolean for internal use only, which is FALSE if computation is done on the same data as for `Jointlcmm` estimation, and TRUE otherwise.

Author(s)

Cecile Proust-Lima and Amadou Diakite

References

Commenges, Liquet and Proust-Lima (2012). Choice of prognostic estimators in joint models by estimating differences of expected conditional Kullback-Leibler risks. Biometrics 68(2), 380-7.

Proust-Lima, Sene, Taylor and Jacqmin-Gadda (2014). Joint latent class models of longitudinal and time-to-event data: a review. Statistical Methods in Medical Research 23, 74-90.

See Also

Jointlcmm,print.epoce,summary.epoce,plot.epoce

Examples

```
## Not run:
## estimation of a joint latent class model with 2 latent classes (ng=2)
# (see the example section of Jointlcmm for details about
#  the model specification)

m <- Jointlcmm(fixed= Ydep1~Time*X1,random=~Time,mixture=~Time,subject='ID'
,survival = Surv(Tevent,Event)~ X1+X2 ,hazard="Weibull"
,hazardtype="PH",ng=2,data=data_lcmm,logscale=TRUE,
B=c(0.7608, -9.4974 , 1.0242,  1.4331 , 0.1063 , 0.6714, 10.4679, 11.3178,
 -2.5671, -0.5386,  1.4616, -0.0605,  0.9489,  0.1020 , 0.2079,  1.5045))
summary(m)

## Computation of the EPOCE on the same dataset as used for
# estimation of m with times at predictions from 1 to 15
VecTime <- c(1,3,5,7,9,11,13,15)
cvpl <- epoce(m,var.time="Time",pred.times=VecTime)
summary(cvpl)
plot(cvpl,bty="l",ylim=c(0,2))

## End(Not run)
```

---

estimates *Maximum likelihood estimates*

Description

This function provides the vector of maximum likelihood estimates of a model estimated with `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`.

Usage

```
## S3 method for class 'hlme'
estimates(x,cholesky=TRUE)
## S3 method for class 'lcmm'
estimates(x,cholesky=TRUE)
## S3 method for class 'Jointlcmm'
estimates(x,cholesky=TRUE)
## S3 method for class 'multlcmm'
estimates(x,cholesky=TRUE)
```

Arguments

`x`	an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`
`cholesky`	optional logical indicating if the parameters of variance-covariance of the random effets should be displayed instead of their cholesky transformations used in the estimation process.

Value

a vector with all estimates of the model.

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

`VarCov`, `hlme`, `lcmm`, `multlcmm`, `Jointlcmm`

---

fitY — *Marginal predictions of the longitudinal outcome(s) in their natural scale from `lcmm`, `Jointlcmm` or `multlcmm` objects*

Description

The function computes the marginal predictions of the longitudinal outcome(s) in their natural scale on the individual data used for the estimation from `lcmm`, `Jointlcmm` or `multlcmm` objects.

Usage

```
## S3 method for class 'lcmm'
fitY(x)
## S3 method for class 'multlcmm'
fitY(x)
## S3 method for class 'Jointlcmm'
fitY(x)
```

Arguments

`x`	an object inheriting from classes `lcmm` or `multlcmm`.

Value

For `lcmm` and `Jointlcmm` objects, returns a matrix with ng+1 columns containing the subject identifier and the ng class-specific marginal predicted values.

For `multlcmm` objects, returns a matrix with ng+2 columns containing the subject identifier, the outcome indicator and the ng class-specific predicted values.

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

`predictY`, `plot.lcmm`

`ForInternalUse` *For internal use only ...*

Description

For internal use only ...

`gridsearch` *Automatic grid search*

Description

This function provides an automatic grid search for latent class mixed models estimated with `hlme`, `lcmm`, `multlcmm` and `Jointlcmm` functions.

Usage

```
gridsearch(m, rep, maxiter, minit)
```

Arguments

`m`	a call of `hlme`, `lcmm`, `multlcmm` or `Jointlcmm` corresponding to the model to estimate
`rep`	the number of departures from random initial values
`maxiter`	the number of iterations in the optimization algorithm
`minit`	an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm` corresponding to the same model as specified in `m` except for the number of classes (it should be one). This object is used to generate random initial values

Details

The function permits the estimation of a model from a grid of random initial values to reduce the odds of a convergence towards a local maximum.

The function was inspired by the emEM technique described in Biernacki et al. (2003). It consists in:

1. randomly generating `rep` sets of initial values for `m` from the estimates of `minit` (this is done internally using option `B=random(minit)` `rep` times)

2. running the optimization algorithm for the model specified in `m` from the `rep` sets of initial values with a maximum number of iterations of `maxit` each time.

3. retaining the estimates of the random initialization that provides the best log-likelihood after `maxiter` iterations.

4. running the optimization algorithm from these estimates for the final estimation.

Value

an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm` corresponding to the call specified in m.

Author(s)

Cecile Proust-Lima and Viviane Philipps

References

Proust-Lima C, Philipps V, Liquet B (2017). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: the R package lcmm. Journal of Statistical Software, 78(2), 1-56

Biernacki C, Celeux G, Govaert G (2003). Choosing Starting Values for the EM Algorithm for Getting the Highest Likelihood in Multivariate Gaussian Mixture models. Computational Statistics and Data Analysis, 41(3-4), 561-575.

Examples

```
## Not run:
# initial model with ng=1 for the random initial values
m1 <- hlme(Y ~ Time * X1, random =~ Time, subject = 'ID', ng = 1,
      data = data_hlme)

# gridsearch with 10 iterations from 50 random departures
m2d <- gridsearch(rep = 50, maxiter = 10, minit = m1, hlme(Y ~ Time * X1,
      mixture =~ Time, random =~ Time, classmb =~ X2 + X3, subject = 'ID',
          ng = 2, data = data_hlme))

## End(Not run)
```

hlme *Estimation of latent class linear mixed models*

Description

This function fits linear mixed models and latent class linear mixed models (LCLMM) also known as growth mixture models or heterogeneous linear mixed models. The LCLMM consists in assuming that the population is divided in a finite number of latent classes. Each latent class is characterised by a specific trajectory modelled by a class-specific linear mixed model. Both the latent class membership and the trajectory can be explained according to covariates. This function is limited to a mixture of Gaussian outcomes. For other types of outcomes, please see function `lcmm`. For multivariate longitudinal outcomes, please see `multlcmm`.

Usage

```
hlme(fixed, mixture, random, subject, classmb, ng = 1,
 idiag = FALSE, nwg = FALSE, cor=NULL, data, B,
 convB=0.0001,convL=0.0001,convG=0.0001,prior,
 maxiter=500, subset=NULL, na.action=1, posfix=NULL,
 verbose=TRUE)
```

Arguments

`fixed` two-sided linear formula object for the fixed-effects in the linear mixed model. The response outcome is on the left of ~ and the covariates are separated by + on the right of ~. By default, an intercept is included. If no intercept, `-1` should be the first term included on the right of ~.

`mixture` one-sided formula object for the class-specific fixed effects in the linear mixed model (to specify only for a number of latent classes greater than 1). Among the list of covariates included in `fixed`, the covariates with class-specific regression parameters are entered in `mixture` separated by `+`. By default, an intercept is included. If no intercept, `-1` should be the first term included.

`random` optional one-sided formula for the random-effects in the linear mixed model. Covariates with a random-effect are separated by `+`. By default, an intercept is included. If no intercept, `-1` should be the first term included.

`subject` name of the covariate representing the grouping structure specified with ''.

`classmb` optional one-sided formula describing the covariates in the class-membership multinomial logistic model. Covariates included are separated by `+`. No intercept should be included in this formula.

`ng` optional number of latent classes considered. If `ng=1` (by default) no `mixture` nor `classmb` should be specified. If `ng>1`, `mixture` is required.

`idiag` optional logical for the structure of the variance-covariance matrix of the random-effects. If `FALSE`, a non structured matrix of variance-covariance is considered (by default). If `TRUE` a diagonal matrix of variance-covariance is considered.

nwg
: optional logical indicating if the variance-covariance of the random-effects is class-specific. If `FALSE` the variance-covariance matrix is common over latent classes (by default). If `TRUE` a class-specific proportional parameter multiplies the variance-covariance matrix in each class (the proportional parameter in the last latent class equals 1 to ensure identifiability).

cor
: optional brownian motion or autoregressive process modeling the correlation between the observations. "BM" or "AR" should be specified, followed by the time variable between brackets. By default, no correlation is added.

data
: optional data frame containing the variables named in `fixed`, `mixture`, `random`, `classmb` and `subject`.

B
: optional specification for the initial values for the parameters. Three options are allowed: (1) a vector of initial values is entered (the order in which the parameters are included is detailed in `details` section). (2) nothing is specified. A preliminary analysis involving the estimation of a standard linear mixed model is performed to choose initial values. (3) when ng>1, a hlme object is entered. It should correspond to the exact same structure of model but with ng=1. The program will automatically generate initial values from this model. This specification avoids the preliminary analysis indicated in (2). Note that due to possible local maxima, the `B` vector should be specified and several different starting points should be tried.

convB
: optional threshold for the convergence criterion based on the parameter stability. By default, convB=0.0001.

convL
: optional threshold for the convergence criterion based on the log-likelihood stability. By default, convL=0.0001.

convG
: optional threshold for the convergence criterion based on the derivatives. By default, convG=0.0001.

prior
: optional name of a covariate containing a prior information about the latent class membership. The covariate should be an integer with values in 0,1,...,ng. Value 0 indicates no prior for the subject while a value in 1,...,ng indicates that the subject belongs to the corresponding latent class.

maxiter
: optional maximum number of iterations for the Marquardt iterative algorithm. By default, maxiter=500.

subset
: a specification of the rows to be used: defaults to all rows. This can be any valid indexing vector for the rows of data or if that is not supplied, a data frame made up of the variable used in formula.

na.action
: Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.

posfix
: Optional vector specifying the indices in vector B of the parameters that should not be estimated. Default to NULL, all parameters are estimated.

verbose
: logical indicating if information about computation should be reported. Default to TRUE.

Details

A. THE VECTOR OF PARAMETERS B

The parameters in the vector of initial values `B` or equivalently in the vector of maximum likelihood estimates `best` are included in the following order:

(1) ng-1 parameters are required for intercepts in the latent class membership model, and when covariates are included in `classmb`, ng-1 paramaters should be entered for each covariate;

(2) for all covariates in `fixed`, one parameter is required if the covariate is not in `mixture`, ng paramaters are required if the covariate is also in `mixture`;

(3) the variance of each random-effect specified in `random` (including the intercept) when `idiag=TRUE`, or the inferior triangular variance-covariance matrix of all the random-effects when `idiag=FALSE`;

(4) only when `nwg=TRUE`, ng-1 parameters are required for the ng-1 class-specific proportional coefficients in the variance covariance matrix of the random-effects;

(5) when `cor` is specified, 1 parameter corresponding to the variance of the Brownian motion should be entered with `cor=BM` and 2 parameters corresponding to the correlation and the variance parameters of the autoregressive process should be entered

(6) the standard error of the residual error.

B. CAUTIONS

Some caution should be made when using the program:

(1) As the log-likelihood of a latent class model can have multiple maxima, a careful choice of the initial values is crucial for ensuring convergence toward the global maximum. The program can be run without entering the vector of initial values (see point 2). However, we recommend to systematically enter initial values in `B` and try different sets of initial values.

(2) The automatic choice of initial values we provide requires the estimation of a preliminary linear mixed model. The user should be aware that first, this preliminary analysis can take time for large datatsets and second, that the generated initial values can be very not likely and even may converge slowly to a local maximum. This is the reason why several alternatives exist. The vector of initial values can be directly specified in `B` the initial values can be generated (automatically or randomly) from a model with `ng=`. Finally, function `gridsearch` performs an automatic grid search.

(3) Convergence criteria are very strict as they are based on the derivatives of the log-likelihood in addition to the parameter stability and log-likelihood stability. In some cases, the program may not converge and reach the maximum number of iterations fixed at 100. In this case, the user should check that parameter estimates at the last iteration are not on the boundaries of the parameter space. If the parameters are on the boundaries of the parameter space, the identifiability of the model is critical. This may happen especially with splines parameters that may be too close to 0 (lower boundary) or classmb parameters that are too high or low (perfect classification). When identifiability of some parameters is suspected, the program can be run again from the former estimates by fixing the suspected parameters to their value with option posfix. This usually solves the problem. An alternative is to remove the parameters of the Beta of Splines link function from the inverse of the Hessian with option partialH. If not, the program should be run again with other initial values, with a higher maximum number of iterations or less strict convergence tolerances.

Value

The list returned is:

ns	number of grouping units in the dataset
ng	number of latent classes
loglik	log-likelihood of the model
best	vector of parameter estimates in the same order as specified in B and detailed in section details
V	vector containing the upper triangle matrix of variance-covariance estimates of Best with exception for variance-covariance parameters of the random-effects for which V contains the variance-covariance estimates of the Cholesky transformed parameters displayed in cholesky
gconv	vector of convergence criteria: 1. on the parameters, 2. on the likelihood, 3. on the derivatives
conv	status of convergence: =1 if the convergence criteria were satisfied, =2 if the maximum number of iterations was reached, =4 or 5 if a problem occured during optimisation
call	the matched call
niter	number of Marquardt iterations
dataset	dataset
N	internal information used in related functions
idiag	internal information used in related functions
pred	table of individual predictions and residuals; it includes marginal predictions (pred_m), marginal residuals (resid_m), subject-specific predictions (pred_ss) and subject-specific residuals (resid_ss) averaged over classes, the observation (obs) and finally the class-specific marginal and subject-specific predictions (with the number of the latent class: pred_m_1,pred_m_2,...,pred_ss_1,pred_ss_2,...)
pprob	table of posterior classification and posterior individual class-membership probabilities
Xnames	list of covariates included in the model
predRE	table containing individual predictions of the random-effects : a column per random-effect, a line per subject
cholesky	vector containing the estimates of the Cholesky transformed parameters of the variance-covariance matrix of the random-effects

Author(s)

Cecile Proust-Lima, Benoit Liquet and Viviane Philipps

<cecile.proust-lima@inserm.fr>

References

Proust-Lima C, Philipps V, Liquet B (2017). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: the R package lcmm. Journal of Statistical Software, 78(2), 1-56

Verbeke G and Lesaffre E (1996). A linear mixed-effects model with heterogeneity in the random-effects population. Journal of the American Statistical Association 91, 217-21

Muthen B and Shedden K (1999). Finite mixture modeling with mixture outcomes using the EM algorithm. Biometrics 55, 463-9

Proust C and Jacqmin-Gadda H (2005). Estimation of linear mixed models with a mixture of distribution for the random-effects. Computer Methods Programs Biomedicine 78, 165-73

See Also

postprob, plot.hlme, summary, predictY

Examples

```
##### Example of a latent class model estimated for a varying number
# of latent classes:
# The model includes a subject- (ID) and class-specific linear
# trend (intercept and Time in fixed, random and mixture components)
# and a common effect of X1 and its interaction with time over classes
# (in fixed).
# The variance of the random intercept and slope are assumed to be equal
# over classes (nwg=F).
# The covariate X3 predicts the class membership (in classmb).
#
# !CAUTION: initialization of mixed models with latent classes is
# of most importance because of the problem of multimodality of the likelihood.
# Calls m2a-m2d illustrate the different implementations for the
# initial values.

### homogeneous linear mixed model (standard linear mixed model)
### with correlated random-effects
m1<-hlme(Y~Time*X1,random=~Time,subject='ID',ng=1,data=data_hlme)
summary(m1)

### latent class linear mixed model with 2 classes

# a. automatic specification from G=1 model estimates:
m2a<-hlme(Y~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,subject='ID',
         ng=2,data=data_hlme,B=m1)

# b. vector of initial values provided by the user:
m2b<-hlme(Y~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,subject='ID',
         ng=2,data=data_hlme,B=c(0.11,-0.74,-0.07,20.71,
                                29.39,-1,0.13,2.45,-0.29,4.5,0.36,0.79,0.97))

# c. random draws from G = 1 model estimates:
m2c<-hlme(Y~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,subject='ID',
          ng=2,data=data_hlme,B=random(m1))

# d. gridsearch with 50 departures and 10 iterations of the algorithm
#     (see function gridsearch for details)
## Not run:
m2d <- gridsearch(rep = 50, maxiter = 10, minit = m1, hlme(Y ~ Time * X1,
mixture =~ Time, random =~ Time, classmb =~ X2 + X3, subject = 'ID', ng = 2,
```

```
data = data_hlme))


## End(Not run)



# summary of the estimation process
summarytable(m1, m2a, m2b, m2c)

# summary of m2a
summary(m2a)

# posterior classification
postprob(m2a)

# plot of predicted trajectories using some newdata
newdata<-data.frame(Time=seq(0,5,length=100),
X1=rep(0,100),X2=rep(0,100),X3=rep(0,100))
plot(predictY(m2a,newdata,var.time="Time"),legend.loc="right",bty="l")
```

Jointlcmm *Estimation of joint latent class models for longitudinal and time-to-event data*

Description

This function fits joint latent class mixed models for a longitudinal outcome and a right-censored (possibly left-truncated) time-to-event. The function handles competing risks and Gaussian or non Gaussian (curvilinear) longitudinal outcomes. For curvilinear longitudinal outcomes, normalizing continuous functions (splines or Beta CDF) can be specified as in `lcmm`.

Usage

```
Jointlcmm(fixed, mixture, random, subject, classmb, ng=1,
 idiag=FALSE, nwg=FALSE, survival,hazard="Weibull",
 hazardtype="Specific", hazardnodes=NULL,TimeDepVar=NULL,
 link=NULL, intnodes=NULL, epsY=0.5, range=NULL,
 cor=NULL,data, B, convB=1e-4, convL=1e-4, convG=1e-4,
 maxiter=100, nsim=100, prior,logscale=FALSE,
 subset=NULL, na.action=1, posfix=NULL, partialH=FALSE,
 verbose=TRUE)
```

Arguments

`fixed` two-sided linear formula object for the fixed-effects in the linear mixed model. The response outcome is on the left of ~ and the covariates are separated by + on the right of the ~. By default, an intercept is included. If no intercept, -1 should be the first term included on the right of ~.

mixture
: one-sided formula object for the class-specific fixed effects in the linear mixed model (to specify only for a number of latent classes greater than 1). Among the list of covariates included in fixed, the covariates with class-specific regression parameters are entered in mixture separated by +. By default, an intercept is included. If no intercept, -1 should be the first term included.

random
: optional one-sided formula for the random-effects in the linear mixed model. Covariates with a random-effect are separated by +. By default, an intercept is included. If no intercept, -1 should be the first term included.

subject
: name of the covariate representing the grouping structure (called subject identifier) specified with ''.

classmb
: optional one-sided formula describing the covariates in the class-membership multinomial logistic model. Covariates included are separated by +. No intercept should be included in this formula.

ng
: optional number of latent classes considered. If ng=1 (by default) no mixture nor classmb should be specified. If ng>1, mixture is required.

idiag
: optional logical for the structure of the variance-covariance matrix of the random-effects. If FALSE, a non structured matrix of variance-covariance is considered (by default). If TRUE a diagonal matrix of variance-covariance is considered.

nwg
: optional logical indicating if the variance-covariance of the random-effects is class-specific. If FALSE the variance-covariance matrix is common over latent classes (by default). If TRUE a class-specific proportional parameter multiplies the variance-covariance matrix in each class (the proportional parameter in the last latent class equals 1 to ensure identifiability).

survival
: two-sided formula object. The left side of the formula corresponds to a surv() object of type "counting" for right-censored and left-truncated data (example: Surv(Time,EntryTime,Indicator)) or of type "right" for right-censored data (example: Surv(Time,Indicator)). Multiple causes of event can be considered in the Indicator (0 for censored, k for cause k of event). The right side of the formula specifies the names of covariates to include in the survival model with mixture() when the effect is class-specific (example: Surv(Time,Indicator) ~ X1 + mixture(X2) for a class-common effect of X1 and a class-specific effect of X2). In the presence of competing events, covariate effects are common by default. Code cause(X3) specifies a cause-specific covariate effect for X3 on each cause of event while cause1(X3) (or cause2(X3), ...) specifies a cause-specific effect of X3 on the first (or second, ...) cause only.

hazard
: optional family of hazard function assumed for the survival model. By default, "Weibull" specifies a Weibull baseline risk function. Other possibilities are "piecewise" for a piecewise constant risk function or "splines" for a cubic M-splines baseline risk function. For these two latter families, the number of nodes and the location of the nodes should be specified as well, separated by -. The number of nodes is entered first followed by -, then the location is specified with "equi", "quant" or "manual" for respectively equidistant nodes, nodes at quantiles of the times of event distribution or interior nodes entered manually in argument hazardnodes. It is followed by - and finally "piecewise" or "splines"

indicates the family of baseline risk function considered. Examples include "5-equi-splines" for M-splines with 5 equidistant nodes, "6-quant-piecewise" for piecewise constant risk over 5 intervals and nodes defined at the quantiles of the times of events distribution and "9-manual-splines" for M-splines risk function with 9 nodes, the vector of 7 interior nodes being entered in the argument `hazardnodes`. In the presence of competing events, a vector of hazards should be provided such as `hazard=c("Weibull","splines"` with 2 causes of event, the first one modelled by a Weibull baseline cause-specific risk function and the second one by splines.

`hazardtype` optional indicator for the type of baseline risk function when ng>1. By default "Specific" indicates a class-specific baseline risk function. Other possibilities are "PH" for a baseline risk function proportional in each latent class, and "Common" for a baseline risk function that is common over classes. In the presence of competing events, a vector of hazardtypes should be given.

`hazardnodes` optional vector containing interior nodes if `splines` or `piecewise` is specified for the baseline hazard function in `hazard`.

`TimeDepVar` optional vector containing an intermediate time corresponding to a change in the risk of event. This time-dependent covariate can only take the form of a time variable with the assumption that there is no effect on the risk before this time and a constant effect on the risk of event after this time (example: initiation of a treatment to account for).

`link` optional family of link functions to estimate. By default, "linear" option specifies a linear link function leading to a standard linear mixed model (homogeneous or heterogeneous as estimated in `hlme`). Other possibilities include "beta" for estimating a link function from the family of Beta cumulative distribution functions, "thresholds" for using a threshold model to describe the correspondence between each level of an ordinal outcome and the underlying latent process, and "Splines" for approximating the link function by I-splines. For this latter case, the number of nodes and the nodes location should be also specified. The number of nodes is first entered followed by `-`, then the location is specified with "equi", "quant" or "manual" for respectively equidistant nodes, nodes at quantiles of the marker distribution or interior nodes entered manually in argument `intnodes`. It is followed by `-` and finally "splines" is indicated. For example, "7-equi-splines" means I-splines with 7 equidistant nodes, "6-quant-splines" means I-splines with 6 nodes located at the quantiles of the marker distribution and "9-manual-splines" means I-splines with 9 nodes, the vector of 7 interior nodes being entered in the argument `intnodes`.

`intnodes` optional vector of interior nodes. This argument is only required for a I-splines link function with nodes entered manually.

`epsY` optional definite positive real used to rescale the marker in (0,1) when the beta link function is used. By default, epsY=0.5.

`range` optional vector indicating the range of the outcome (that is the minimum and maximum). By default, the range is defined according to the minimum and maximum observed values of the outcome. The option should be used only for Beta and Splines transformations.

`cor` optional brownian motion or autoregressive process modeling the correlation between the observations. "BM" or "AR" should be specified, followed by the time variable between brackets. By default, no correlation is added.

`data` optional data frame containing the variables named in `fixed`, `mixture`, `random`, `classmb` and `subject`.

`B` optional specification for the initial values for the parameters. Three options are allowed: (1) a vector of initial values is entered (the order in which the parameters are included is detailed in `details` section). (2) nothing is specified. A preliminary analysis involving the estimation of a standard linear mixed model is performed to choose initial values. (3) when ng>1, a multlcmm object is entered. It should correspond to the exact same structure of model but with ng=1. The program will automatically generate initial values from this model. This specification avoids the preliminary analysis indicated in (2) Note that due to possible local maxima, the `B` vector should be specified and several different starting points should be tried.

`convB` optional threshold for the convergence criterion based on the parameter stability. By default, convB=0.0001.

`convL` optional threshold for the convergence criterion based on the log-likelihood stability. By default, convL=0.0001.

`convG` optional threshold for the convergence criterion based on the derivatives. By default, convG=0.0001.

`maxiter` optional maximum number of iterations for the Marquardt iterative algorithm. By default, maxiter=150.

`nsim` optional number of points for the predicted survival curves and predicted baseline risk curves. By default, nsim=100.

`prior` optional name of a covariate containing a prior information about the latent class membership. The covariate should be an integer with values in 0,1,...,ng. Value O indicates no prior for the subject while a value in 1,...,ng indicates that the subject belongs to the corresponding latent class.

`logscale` optional boolean indicating whether an exponential (logscale=TRUE) or a square (logscale=FALSE -by default) transformation is used to ensure positivity of parameters in the baseline risk functions. See details section

`subset` a specification of the rows to be used: defaults to all rows. This can be any valid indexing vector for the rows of data or if that is not supplied, a data frame made up of the variable used in formula.

`na.action` Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.

`posfix` Optional vector specifying the indices in vector B of the parameters that should not be estimated. Default to NULL, all parameters are estimated.

`partialH` optional logical for Piecewise and Splines baseline risk functions only. Indicates whether the parameters of the baseline risk functions can be dropped from the Hessian matrix to define convergence criteria.

`verbose` logical indicating if information about computation should be reported. Default to TRUE.

Details

A. BASELINE RISK FUNCTIONS

For the baseline risk functions, the following parameterizations were considered. Be careful, parametrisations changed in lcmm_V1.5:

1. With the "Weibull" function: 2 parameters are necessary w_1 and w_2 so that the baseline risk function a_0(t) = w_1^2*w_2^2*(w_1^2*t)^(w_2^2-1) if logscale=FALSE and a_0(t) = exp(w_1)*exp(w_2)(t)^(exp(w_2)-1) if logscale=TRUE.

2. with the "piecewise" step function and nz nodes (y_1,...y_nz), nz-1 parameters are necesssary p_1,...p_nz-1 so that the baseline risk function a_0(t) = p_j^2 for y_j < t =< y_j+1 if logscale=FALSE and a_0(t) = exp(p_j) for y_j < t =< y_j+1 if logscale=TRUE.

3. with the "splines" function and nz nodes (y_1,...y_nz), nz+2 parameters are necessary s_1,...s_nz+2 so that the baseline risk function a_0(t) = sum_j s_j^2 M_j(t) if logscale=FALSE and a_0(t) = sum_j exp(s_j) M_j(t) if logscale=TRUE where {M_j} is the basis of cubic M-splines.

Two parametrizations of the baseline risk function are proposed (logscale=TRUE or FALSE) because in some cases, especially when the instantaneous risks are very close to 0, some convergence problems may appear with one parameterization or the other. As a consequence, we recommend to try the alternative parameterization (changing logscale option) when a joint latent class model does not converge (maximum number of iterations reached) where as convergence criteria based on the parameters and likelihood are small.

B. THE VECTOR OF PARAMETERS B

The parameters in the vector of initial values `B` or in the vector of maximum likelihood estimates `best` are included in the following order: (1) ng-1 parameters are required for intercepts in the latent class membership model, and if covariates are included in `classmb`, ng-1 parameters should be entered for each one; (2) parameters for the baseline risk function: 2 parameters for each Weibull, nz-1 for each piecewise constant risk and nz+2 for each splines risk; this number should be multiplied by ng if specific hazard is specified; otherwise, ng-1 additional proportional effects are expected if PH hazard is specified; otherwise nothing is added if common hazard is specified. In the presence of competing events, the number of parameters should be adapted to the number of causes of event; (3) for all covariates in `survival`, ng parameters are required if the covariate is inside a `mixture()`, otherwise 1 parameter is required. Covariates parameters should be included in the same order as in `survival`. In the presence of cause-specific effects, the number of parameters should be multiplied by the number of causes; (4) for all covariates in `fixed`, one parameter is required if the covariate is not in `mixture`, ng parameters are required if the covariate is also in `mixture`. Parameters should be included in the same order as in `fixed`; (5) the variance of each random-effect specified in `random` (including the intercept) if `idiag=TRUE` and the inferior triangular variance-covariance matrix of all the random-effects if `idiag=FALSE`; (6) only if `nwg=TRUE`, ng-1 parameters for class-specific proportional coefficients for the variance covariance matrix of the random-effects; (7) the variance of the residual error.

C. CAUTION

Some caution should be made when using the program:

(1) As the log-likelihood of a latent class model can have multiple maxima, a careful choice of the initial values is crucial for ensuring convergence toward the global maximum. The program can be run without entering the vector of initial values (see point 2). However, we recommend to systematically enter initial values in `B` and try different sets of initial values.

(2) The automatic choice of initial values that we provide requires the estimation of a preliminary linear mixed model. The user should be aware that first, this preliminary analysis can take time for large datatsets and second, that the generated initial values can be very not likely and even may converge slowly to a local maximum. This is a reason why several alternatives exist. The vector of initial values can be directly specified in `B` the initial values can be generated (automatically or randomly) from a model with `ng=`. Finally, function `gridsearch` performs an automatic grid search.

(3) Convergence criteria are very strict as they are based on derivatives of the log-likelihood in addition to the parameter and log-likelihood stability. In some cases, the program may not converge and reach the maximum number of iterations fixed at 150. In this case, the user should check that parameter estimates at the last iteration are not on the boundaries of the parameter space. If the parameters are on the boundaries of the parameter space, the identifiability of the model is critical. This may happen especially when baseline risk functions involve splines (value close to the lower boundary - 0 with logscale=F -infinity with logscale=F) or classmb parameters that are too high or low (perfect classification) or linkfunction parameters. When identifiability of some parameters is suspected, the program can be run again from the former estimates by fixing the suspected parameters to their value with option posfix. This usually solves the problem. An alternative is to remove the parameters of the Beta of Splines link function from the inverse of the Hessian with option partialH. If not, the program should be run again with other initial values. Some problems of convergence may happen when the instantaneous risks of event are very low and "piecewise" or "splines" baseline risk functions are specified. In this case, changing the parameterization of the baseline risk functions with option logscale is recommended (see paragraph A for details).

Value

The list returned is:

`loglik`	log-likelihood of the model
`best`	vector of parameter estimates in the same order as specified in `B` and detailed in section `details`
`V`	vector containing the upper triangle matrix of variance-covariance estimates of `Best` with exception for variance-covariance parameters of the random-effects for which `V` contains the variance-covariance estimates of the Cholesky transformed parameters displayed in `cholesky`
`gconv`	vector of convergence criteria: 1. on the parameters, 2. on the likelihood, 3. on the derivatives
`conv`	status of convergence: =1 if the convergence criteria were satisfied, =2 if the maximum number of iterations was reached, =4 or 5 if a problem occured during optimisation
`call`	the matched call
`niter`	number of Marquardt iterations
`pred`	table of individual predictions and residuals; it includes marginal predictions (pred_m), marginal residuals (resid_m), subject-specific predictions (pred_ss) and subject-specific residuals (resid_ss) averaged over classes, the observation (obs) and finally the class-specific marginal and subject-specific predictions (with the number of the latent class: pred_m_1,pred_m_2,...,pred_ss_1,pred_ss_2,...)

`pprob`	table of posterior classification and posterior individual class-membership probabilities based on the longitudinal data and the time-to-event data
`pprobY`	table of posterior classification and posterior individual class-membership probabilities based only on the longitudinal data
`predRE`	table containing individual predictions of the random-effects: a column per random-effect, a line per subject
`cholesky`	vector containing the estimates of the Cholesky transformed parameters of the variance-covariance matrix of the random-effects
`scoretest`	Statistic of the Score Test for the conditional independence assumption of the longitudinal and survival data given the latent class structure. Under the null hypothesis, the statistics is a Chi-square with p degrees of freedom where p indicates the number of random-effects in the longitudinal mixed model. See Jacqmin-Gadda and Proust-Lima (2009) for more details.
`predSurv`	table of predictions giving for the window of times to event (called "time"), the predicted baseline risk function in each latent class (called "RiskFct") and the predicted cumulative baseline risk function in each latent class (called "CumRiskFct").
`hazard`	internal information about the hazard specification used in related functions

Author(s)

Cecile Proust Lima, Amadou Diakite and Viviane Philipps

`<cecile.proust-lima@inserm.fr>`

References

Proust-Lima C, Philipps V, Liquet B (2017). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: the R package lcmm. Journal of Statistical Software, 78(2), 1-56

Lin, H., Turnbull, B. W., McCulloch, C. E. and Slate, E. H. (2002). Latent class models for joint analysis of longitudinal biomarker and event process data: application to longitudinal prostate-specific antigen readings and prostate cancer. Journal of the American Statistical Association 97, 53-65.

Proust-Lima, C. and Taylor, J. (2009). Development and validation of a dynamic prognostic tool for prostate cancer recurrence using repeated measures of post-treatment PSA: a joint modelling approach. Biostatistics 10, 535-49.

Jacqmin-Gadda, H. and Proust-Lima, C. (2010). Score test for conditional independence between longitudinal outcome and time-to-event given the classes in the joint latent class model. Biometrics 66(1), 11-9

Proust-Lima, Sene, Taylor and Jacqmin-Gadda (2014). Joint latent class models of longitudinal and time-to-event data: a review. Statistical Methods in Medical Research 23, 74-90.

See Also

`postprob`, `plot.Jointlcmm`, `plot.predict`, `epoce`

Examples

```
#### Example of a joint latent class model estimated for a varying number
# of latent classes:
# The linear mixed model includes a subject- (ID) and class-specific
# linear trend (intercept and Time in fixed, random and mixture components)
# and a common effect of X1 and its interaction with time over classes
# (in fixed).
# The variance of the random intercept and slopes are assumed to be equal
# over classes (nwg=F).
# The covariate X3 predicts the class membership (in classmb).
# The baseline hazard function is modelled with cubic M-splines -3
# nodes at the quantiles- (in hazard) and a proportional hazard over
# classes is assumed (in hazardtype). Covariates X1 and X2 predict the
# risk of event (in survival) with a common effect over classes for X1
# and a class-specific effect of X2.
# !CAUTION: for illustration, only default initial values where used but
# other sets of initial values should be tried to ensure convergence
# towards the global maximum.


## Not run:
#### estimation with 1 latent class (ng=1): independent models for the
# longitudinal outcome and the time of event
m1 <- Jointlcmm(fixed= Ydep1~X1*Time,random=~Time,subject='ID'
,survival = Surv(Tevent,Event)~ X1+X2 ,hazard="3-quant-splines"
,hazardtype="PH",ng=1,data=data_lcmm)
summary(m1)
#Goodness-of-fit statistics for m1:
#    maximum log-likelihood: -3944.77 ; AIC: 7919.54  ;  BIC: 7975.09

## End(Not run)

#### estimation with 2 latent classes (ng=2)
m2 <- Jointlcmm(fixed= Ydep1~Time*X1,mixture=~Time,random=~Time,
classmb=~X3,subject='ID',survival = Surv(Tevent,Event)~X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",ng=2,data=data_lcmm,
B=c(0.64,-0.62,0,0,0.52,0.81,0.41,0.78,0.1,0.77,-0.05,10.43,11.3,-2.6,
-0.52,1.41,-0.05,0.91,0.05,0.21,1.5))
summary(m2)
#Goodness-of-fit statistics for m2:
#       maximum log-likelihood: -3921.27; AIC: 7884.54; BIC: 7962.32

## Not run:
#### estimation with 3 latent classes (ng=3)
m3 <- Jointlcmm(fixed= Ydep1~Time*X1,mixture=~Time,random=~Time,
classmb=~X3,subject='ID',survival = Surv(Tevent,Event)~ X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",ng=3,data=data_lcmm,
B=c(0.77,0.4,-0.82,-0.27,0,0,0,0.3,0.62,2.62,5.31,-0.03,1.36,0.82,
-13.5,10.17,10.24,11.51,-2.62,-0.43,-0.61,1.47,-0.04,0.85,0.04,0.26,1.5))
summary(m3)
```

```
#Goodness-of-fit statistics for m3:
#       maximum log-likelihood: -3890.26 ; AIC: 7834.53;  BIC: 7934.53

#### estimation with 4 latent classes (ng=4)
m4 <- Jointlcmm(fixed= Ydep1~Time*X1,mixture=~Time,random=~Time,
classmb=~X3,subject='ID',survival = Surv(Tevent,Event)~ X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",ng=4,data=data_lcmm,
B=c(0.54,-0.42,0.36,-0.94,-0.64,-0.28,0,0,0,0.34,0.59,2.6,2.56,5.26,
-0.1,1.27,1.34,0.7,-5.72,10.54,9.02,10.2,11.58,-2.47,-2.78,-0.28,-0.57,
1.48,-0.06,0.61,-0.07,0.31,1.5))
summary(m4)
#Goodness-of-fit statistics for m4:
#   maximum log-likelihood: -3886.93 ; AIC: 7839.86;  BIC: 7962.09


##### The model with 3 latent classes is retained according to the BIC
##### and the conditional independence assumption is not rejected at
##### the 5% level.
# posterior classification
plot(m3,which="postprob")
# Class-specific predicted baseline risk & survival functions in the
# 3-class model retained (for the reference value of the covariates)
plot(m3,which="baselinerisk",bty="l")
plot(m3,which="baselinerisk",ylim=c(0,5),bty="l")
plot(m3,which="survival",bty="l")
# class-specific predicted trajectories in the 3-class model retained
# (with characteristics of subject ID=193)
data <- data_lcmm[data_lcmm$ID==193,]
plot(predictY(m3,var.time="Time",newdata=data,bty="l")
# predictive accuracy of the model evaluated with EPOCE
vect <- 1:15
cvpl <- epoce(m3,var.time="Time",pred.times=vect)
summary(cvpl)
plot(cvpl,bty="l",ylim=c(0,2))
############## end of example ##############

## End(Not run)
```

`lcmm` *Estimation of mixed-effect models and latent class mixed-effect models for different types of outcomes (continuous Gaussian, continuous non-Gaussian or ordinal)*

Description

This function fits mixed models and latent class mixed models for different types of outcomes. It handles continuous longitudinal outcomes (Gaussian or non-Gaussian) as well as bounded quantitative, discrete and ordinal longitudinal outcomes. The different types of outcomes are taken into account using parameterized nonlinear link functions between the observed outcome and the underlying latent process of interest it measures. At the latent process level, the model estimates a standard linear mixed model or a latent class linear mixed model when heterogeneity in the population is investigated (in the same way as in function hlme). It should be noted that the program also works when no random-effect is included. Parameters of the nonlinear link function and of the latent process mixed model are estimated simultaneously using a maximum likelihood method.

Usage

```
lcmm(fixed, mixture, random, subject, classmb, ng = 1,
 idiag = FALSE, nwg = FALSE, link = "linear", intnodes = NULL,
 epsY = 0.5, cor=NULL, data, B, convB = 1e-04, convL = 1e-04,
 convG = 1e-04, maxiter=100, nsim=100, prior,range=NULL,
 subset=NULL, na.action=1, posfix=NULL, partialH=FALSE,
 verbose=TRUE)
```

Arguments

fixed	a two-sided linear formula object for specifying the fixed-effects in the linear mixed model at the latent process level. The response outcome is on the left of ~ and the covariates are separated by + on the right of the ~. Fo identifiability purposes, the intercept specified by default should not be removed by a -1.
mixture	a one-sided formula object for the class-specific fixed effects in the latent process mixed model (to specify only for a number of latent classes greater than 1). Among the list of covariates included in fixed, the covariates with class-specific regression parameters are entered in mixture separated by +. By default, an intercept is included. If no intercept, -1 should be the first term included.
random	an optional one-sided formula for the random-effects in the latent process mixed model. Covariates with a random-effect are separated by +. By default, an intercept is included. If no intercept, -1 should be the first term included.
subject	name of the covariate representing the grouping structure.
classmb	an optional one-sided formula describing the covariates in the class-membership multinomial logistic model. Covariates included are separated by +. No intercept should be included in this formula.
ng	number of latent classes considered. If ng=1 no mixture nor classmb should be specified. If ng>1, mixture is required.
idiag	optional logical for the variance-covariance structure of the random-effects. If FALSE, a non structured matrix of variance-covariance is considered (by default). If TRUE a diagonal matrix of variance-covariance is considered.
nwg	optional logical of class-specific variance-covariance of the random-effects. If FALSE the variance-covariance matrix is common over latent classes (by default). If TRUE a class-specific proportional parameter multiplies the variance-covariance matrix in each class (the proportional parameter in the last latent class equals 1 to ensure identifiability).
link	optional family of link functions to estimate. By default, "linear" option specifies a linear link function leading to a standard linear mixed model (homogeneous or heterogeneous as estimated in hlme). Other possibilities include "beta" for estimating a link function from the family of Beta cumulative distribution

functions, "thresholds" for using a threshold model to describe the correspondence between each level of an ordinal outcome and the underlying latent process, and "Splines" for approximating the link function by I-splines. For this latter case, the number of nodes and the nodes location should be also specified. The number of nodes is first entered followed by `-`, then the location is specified with "equi", "quant" or "manual" for respectively equidistant nodes, nodes at quantiles of the marker distribution or interior nodes entered manually in argument `intnodes`. It is followed by `-` and finally "splines" is indicated. For example, "7-equi-splines" means I-splines with 7 equidistant nodes, "6-quant-splines" means I-splines with 6 nodes located at the quantiles of the marker distribution and "9-manual-splines" means I-splines with 9 nodes, the vector of 7 interior nodes being entered in the argument `intnodes`.

`intnodes` optional vector of interior nodes. This argument is only required for a I-splines link function with nodes entered manually.

`epsY` optional definite positive real used to rescale the marker in (0,1) when the beta link function is used. By default, epsY=0.5.

`cor` optional brownian motion or autoregressive process modeling the correlation between the observations. "BM" or "AR" should be specified, followed by the time variable between brackets. By default, no correlation is added.

`data` optional data frame containing the variables named in `fixed`, `mixture`, `random`, `classmb` and `subject`.

`B` optional specification for the initial values for the parameters. Three options are allowed: (1) a vector of initial values is entered (the order in which the parameters are included is detailed in `details` section). (2) nothing is specified. A preliminary analysis involving the estimation of a standard linear mixed model is performed to choose initial values. (3) when ng>1, a lcmm object is entered. It should correspond to the exact same structure of model but with ng=1. The program will automatically generate initial values from this model. This specification avoids the preliminary analysis indicated in (2). Note that due to possible local maxima, the `B` vector should be specified and several different starting points should be tried.

`convB` optional threshold for the convergence criterion based on the parameter stability. By default, convB=0.0001.

`convL` optional threshold for the convergence criterion based on the log-likelihood stability. By default, convL=0.0001.

`convG` optional threshold for the convergence criterion based on the derivatives. By default, convG=0.0001.

`maxiter` optional maximum number of iterations for the Marquardt iterative algorithm. By default, maxiter=100.

`nsim` number of points used to plot the estimated link function. By default, nsim=100.

`prior` name of the covariate containing the prior on the latent class membership. The covariate should be an integer with values in 0,1,...,ng. When there is no prior, the value should be 0. When there is a prior for the subject, the value should be the number of the latent class (in 1,...,ng).

`range` optional vector indicating the range of the outcome (that is the minimum and maximum). By default, the range is defined according to the minimum and maximum observed values of the outcome. The option should be used only for Beta and Splines transformations.

`subset` optional vector giving the subset of observations in `data` to use. By default, all lines.

`na.action` Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.

`posfix` Optional vector specifying the indices in vector B of the parameters that should not be estimated. Default to NULL, all parameters are estimated.

`partialH` optional logical for Beta or Splines link functions only. Indicates whether the parameters of the link functions can be dropped from the Hessian matrix to define convergence criteria.

`verbose` logical indicating if information about computation should be reported. Default to TRUE.

Details

A. THE PARAMETERIZED LINK FUNCTIONS

`lcmm` function estimates mixed models and latent class mixed models for different types of outcomes by assuming a parameterized link function for linking the outcome Y(t) with the underlying latent process L(t) it measures. To fix the latent process dimension, we chose to constrain the (first) intercept of the latent class mixed model at the latent process level at 0 and the standard error of the gaussian error of measurement at 1. These two parameters are replaced by additional parameters in the parameterized link function :

1. With the "linear" link function, 2 parameters are required that correspond directly to the intercept and the standard error: (Y - b1)/b2 = L(t).

2. With the "beta" link function, 4 parameters are required for the following transformation: [h(Y(t)',b1,b2) - b3]/b4 where h is the Beta CDF with canonical parameters c1 and c2 that can be derived from b1 and b2 as c1=exp(b1)/[exp(b2)*(1+exp(b1))] and c2=1/[exp(b2)*(1+exp(b1))], and Y(t)' is the rescaled outcome i.e. Y(t)'= [Y(t) - min(Y(t)) + epsY] / [max(Y(t)) - min(Y(t)) +2*epsY].

3. With the "splines" link function, n+2 parameters are required for the following transformation b_1 + b_2*I_1(Y(t)) + ... + b_{n+2} I_{n+1}(Y(t)), where I_1,...,I_{n+1} is the basis of quadratic I-splines. To constraint the parameters to be positive, except for b_1, the program estimates b_k^* (for k=2,...,n+2) so that b_k=(b_k^*)^2.

4. With the "thresholds" link function for an ordinal outcome in levels 0,...,C. A maximumn of C parameters are required for the following transformation: Y(t)=c <=> b_c < L(t) <= b_{c+1} with b_0 = - infinity and b_{C+1}=+infinity. The number of parameters is reduced if some levels do not have any information. For example, if a level c is not observed in the dataset, the corresponding threshold b_{c+1} is constrained to be the same as the previous one b_{c}. The number of parameters in the link function is reduced by 1.

To constraint the parameters to be increasing, except for the first parameter b_1, the program estimates b_k^* (for k=2,...C) so that b_{k}=b_{k-1}+(b_k^*)^2.

Details of these parameterized link functions can be found in the referred papers.

B. THE VECTOR OF PARAMETERS B

The parameters in the vector of initial values `B` or in the vector of maximum likelihood estimates `best` are included in the following order: (1) ng-1 parameters are required for intercepts in the latent class membership model, and if covariates are included in `classmb`, ng-1 paramaters should be entered for each one; (2) for all covariates in `fixed`, one parameter is required if the covariate is not in `mixture`, ng paramaters are required if the covariate is also in `mixture`; When ng=1, the intercept is not estimated and no parameter should be specified in `B`. When ng>1, the first intercept is not estimated and only ng-1 parameters should be specified in `B`; (3) the variance of each random-effect specified in `random` (including the intercept) if `idiag=TRUE` and the inferior triangular variance-covariance matrix of all the random-effects if `idiag=FALSE`; (4) only if `nwg=TRUE`, ng-1 parameters for class-specific proportional coefficients for the variance covariance matrix of the random-effects; (5) In contrast with hlme, due to identifiability purposes, the standard error of the Gaussian error is not estimated (fixed at 1), and should not be specified in `B`; (6) The parameters of the link function: 2 for "linear", 4 for "beta", n+2 for "splines" with n nodes and the number of levels minus one for "thresholds".

C. CAUTIONS REGARDING THE USE OF THE PROGRAM

Some caution should be made when using the program. convergence criteria are very strict as they are based on derivatives of the log-likelihood in addition to the parameter and log-likelihood stability. In some cases, the program may not converge and reach the maximum number of iterations fixed at 100. In this case, the user should check that parameter estimates at the last iteration are not on the boundaries of the parameter space. If the parameters are on the boundaries of the parameter space, the identifiability of the model is critical. This may happen especially with splines parameters that may be too close to 0 (lower boundary) or classmb parameters that are too high or low (perfect classification). When identifiability of some parameters is suspected, the program can be run again from the former estimates by fixing the suspected parameters to their value with option posfix. This usually solves the problem. An alternative is to remove the parameters of the Beta of Splines link function from the inverse of the Hessian with option partialH. If not, the program should be run again with other initial values, with a higher maximum number of iterations or less strict convergence tolerances.

Specifically when investigating heterogeneity (that is with ng>1): (1) As the log-likelihood of a latent class model can have multiple maxima, a careful choice of the initial values is crucial for ensuring convergence toward the global maximum. The program can be run without entering the vector of initial values (see point 2). However, we recommend to systematically enter initial values in `B` and try different sets of initial values. (2) The automatic choice of initial values we provide requires the estimation of a preliminary linear mixed model. The user should be aware that first, this preliminary analysis can take time for large datatsets and second, that the generated initial values can be very not likely and even may converge slowly to a local maximum. This is the reason why several alternatives exist. The vector of initial values can be directly specified in `B` the initial values can be generated (automatically or randomly) from a model with `ng=`. Finally, function `gridsearch` performs an automatic grid search.

D. NUMERICAL INTEGRATION WITH THE THRESHOLD LINK FUNCTION

With exception for the threshold link function, maximum likelihood estimation implemented in lcmm does not require any numerical integration over the random-effects so that the estimation procedure is relatively fast. See Proust et al. (2006) for more details on the estimation procedure.

However, with the threshold link function and when at least one random-effect is specified, a numerical integration over the random-effects distribution is required in each computation of the individual contribution to the likelihood which complicates greatly the estimation procedure. For the moment, we do not allow any option regarding the numerical integration technics used. 1. When a single random-effect is specified, we use a standard non-adaptive Gaussian quadrature with 30 points. 2. When at least two random-effects are specified, we use a multivariate non-adaptive Gaussian quadrature implemented by Genz (1996) in HRMSYM Fortran subroutine.

Further developments should allow for adaptive technics and more options regarding the numerical integration technic.

E. POSTERIOR DISCRETE LIKELIHOOD

Models involving nonlinear continuous link functions assume the continuous data while the model with a threshold model assumes discrete data. As a consequence, comparing likelihoods or criteria based on the likelihood (as AIC) for these models is not possible as the former are based on a Lebesgue measure and the latter on a counting measure. To make the comparison possible, we compute the posterior discrete likelihood for all the models with a nonlinear continuous link function. This posterior likelihood considers the data as discrete; it is computed at the MLE (maximum likelihood estimates) using the counting measure so that models with threshold or continuous link functions become comparable. Further details can be found in Proust-Lima, Amieva, Jacqmin-Gadda (2012).

In addition to the Akaike information criterion based on the discrete posterior likelihood, we also compute a universal approximate cross-validation criterion to compare models based on a different measure. See Commenges, Proust-Lima, Samieri, Liquet (2015) for further details.

Value

The list returned is:

`ns`	number of grouping units in the dataset
`ng`	number of latent classes
`loglik`	log-likelihood of the model
`best`	vector of parameter estimates in the same order as specified in `B` and detailed in section `details`
`V`	vector containing the upper triangle matrix of variance-covariance estimates of `Best` with exception for variance-covariance parameters of the random-effects for which `V` contains the variance-covariance estimates of the Cholesky transformed parameters displayed in `cholesky`
`gconv`	vector of convergence criteria: 1. on the parameters, 2. on the likelihood, 3. on the derivatives
`conv`	status of convergence: =1 if the convergence criteria were satisfied, =2 if the maximum number of iterations was reached, =4 or 5 if a problem occured during optimisation
`call`	the matched call
`niter`	number of Marquardt iterations
`dataset`	dataset
`N`	internal information used in related functions

`idiag`	internal information used in related functions
`pred`	table of individual predictions and residuals in the underlying latent process scale; it includes marginal predictions (pred_m), marginal residuals (resid_m), subject-specific predictions (pred_ss) and subject-specific residuals (resid_ss) averaged over classes, the transformed observations in the latent process scale (obs) and finally the class-specific marginal and subject-specific predictions (with the number of the latent class: pred_m_1,pred_m_2,...,pred_ss_1,pred_ss_2,...). This output is not available yet when specifying a thresholds transformation.
`pprob`	table of posterior classification and posterior individual class-membership probabilities
`Xnames`	list of covariates included in the model
`predRE`	table containing individual predictions of the random-effects : a column per random-effect, a line per subject. This output is not available yet when specifying a thresholds transformation.
`cholesky`	vector containing the estimates of the Cholesky transformed parameters of the variance-covariance matrix of the random-effects
`estimlink`	table containing the simulated values of the marker and corresponding estimated link function
`epsY`	definite positive real used to rescale the marker in (0,1) when the beta link function is used. By default, epsY=0.5.
`linktype`	indicator of link function type: 0 for linear, 1 for beta, 2 for splines and 3 for thresholds
`linknodes`	vector of nodes useful only for the 'splines' link function

Author(s)

Cecile Proust-Lima, Amadou Diakite, Benoit Liquet and Viviane Philipps

`<cecile.proust-lima@inserm.fr>`

References

Proust-Lima C, Philipps V, Liquet B (2017). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: the R package lcmm. Journal of Statistical Software, 78(2), 1-56

Genz and Keister (1996). Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight. Journal of Computational and Applied Mathematics 71: 299-309.

Proust and Jacqmin-Gadda (2005). Estimation of linear mixed models with a mixture of distribution for the random-effects. Comput Methods Programs Biomed 78: 165-73.

Proust, Jacqmin-Gadda, Taylor, Ganiayre, and Commenges (2006). A nonlinear model with latent process for cognitive evolution using multivariate longitudinal data. Biometrics 62: 1014-24.

Proust-Lima, Dartigues and Jacqmin-Gadda (2011). Misuse of the linear mixed model when evaluating risk factors of cognitive decline. Amer J Epidemiol 174(9): 1077-88.

Proust-Lima, Amieva and Jacqmin-Gadda (2013). Analysis of multivariate mixed longitudinal data : a flexible latent process approach, British Journal of Mathematical and Statistical Psychology 66(3): 470-87.

Commenges, Proust-Lima, Samieri, Liquet (2015). A universal approximate cross-validation criterion for regular risk functions. Int J Biostat. 2015 May;11(1):51-67

See Also

postprob, plot.lcmm, plot.predict, hlme

Examples

```
## Not run:
#### Estimation of homogeneous mixed models with different assumed link
#### functions, a quadratic mean trajectory for the latent process and
#### correlated random intercept and slope (the random quadratic slope
#### was removed as it did not improve the fit of the data).
#### -- comparison of linear, Beta and 3 different splines link functions --
# linear link function
m10<-lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="linear")
summary(m10)
# Beta link function
m11<-lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="beta")
summary(m11)
plot(m11,which="linkfunction",bty="l")
# I-splines with 3 equidistant nodes
m12<-lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="3-equi-splines")
summary(m12)
# I-splines with 5 nodes at quantiles
m13<-lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="5-quant-splines")
summary(m13)
# I-splines with 5 nodes, and interior nodes entered manually
m14<-lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="5-manual-splines",intnodes=c(10,20,25))
summary(m14)
plot(m14,which="linkfunction",bty="l")


# Thresholds
# Especially for the threshold link function, we recommend to estimate
# models with increasing complexity and use estimates of previous ones
# to specify plausible initial values (we remind that estimation of
# models with threshold link function involves a computationally demanding
# numerical integration -here of size 3)
m15<-lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1
,data=data_lcmm,link="thresholds",maxiter=100,
B=c(-0.8379, -0.1103,  0.3832,  0.3788 , 0.4524, -7.3180,  0.5917,  0.7364,
 0.6530, 0.4038,  0.4290,  0.6099,  0.6014 , 0.5354 , 0.5029 , 0.5463,
 0.5310 , 0.5352, 0.6498,  0.6653,  0.5851,  0.6525,  0.6701 , 0.6670 ,
 0.6767 , 0.7394 , 0.7426, 0.7153,  0.7702,  0.6421))
summary(m15)
plot(m15,which="linkfunction",bty="l")

#### Plot of estimated different link functions:
#### (applicable for models that only differ in the "link function" used.
```

```
####  Otherwise, the latent process scale is different and a rescaling
####  is necessary)
plot(m10,which="linkfunction",col=1,xlab="latent process",ylab="marker",
bty="l",xlim=c(-10,5),legend=NULL)
plot(m11,which="linkfunction",add=TRUE,col=2,legend=NULL)
plot(m12,which="linkfunction",add=TRUE,col=3,legend=NULL)
plot(m13,which="linkfunction",add=TRUE,col=4,legend=NULL)
plot(m14,which="linkfunction",add=TRUE,col=5,legend=NULL)
plot(m15,which="linkfunction",add=TRUE,col=6,legend=NULL)
legend(x="bottomright",legend=c("linear","beta","spl_3e","spl_5q","spl_5m","thresholds"),
col=1:6,lty=1,inset=.02,box.lty=0)

#### Estimation of 2-latent class mixed models with different assumed link
#### functions with individual and class specific linear trend
#### for illustration, only default initial values where used but other
#### sets of initial values should also be tried to ensure convergence
#### towards the golbal maximum
# Linear link function
m20<-lcmm(Ydep2~Time,random=~Time,subject='ID',mixture=~Time,ng=2,
idiag=TRUE,data=data_lcmm,link="linear",B=c(-0.98,0.79,-2.09,
-0.81,0.19,0.55,24.49,2.24))
summary(m20)
postprob(m20)
# Beta link function
m21<-lcmm(Ydep2~Time,random=~Time,subject='ID',mixture=~Time,ng=2,
idiag=TRUE,data=data_lcmm,link="beta",B=c(-0.1,-0.56,-0.4,-1.77,
0.53,0.14,0.6,-0.83,0.73,0.09))
summary(m21)
postprob(m21)
# I-splines link function (and 5 nodes at quantiles)
m22<-lcmm(Ydep2~Time,random=~Time,subject='ID',mixture=~Time,ng=2,
idiag=TRUE,data=data_lcmm,link="5-quant-splines",B=c(0.12,0.63,
-1.76,-0.39,0.51,0.13,-7.37,1.05,1.28,1.96,1.3,0.93,1.05))
summary(m22)
postprob(m22)

data <- data_lcmm[data_lcmm$ID==193,]
plot(predictL(m22,var.time="Time",newdata=data,bty="l")


## End(Not run)
```

---

methods *Standard methods for estimated models*

Description

coef, vcov, fixef, ranef, fitted and residuals methods for estimated hlme, lcmm, mutlcmm and Jointlcmm models.

Usage

```
## S3 method for class 'hlme'
coef(object,...)
## S3 method for class 'lcmm'
coef(object,...)
## S3 method for class 'Jointlcmm'
coef(object,...)
## S3 method for class 'multlcmm'
coef(object,...)

## S3 method for class 'hlme'
vcov(object,...)
## S3 method for class 'lcmm'
vcov(object,...)
## S3 method for class 'Jointlcmm'
vcov(object,...)
## S3 method for class 'multlcmm'
vcov(object,...)

## S3 method for class 'hlme'
fixef(object,...)
## S3 method for class 'lcmm'
fixef(object,...)
## S3 method for class 'Jointlcmm'
fixef(object,...)
## S3 method for class 'multlcmm'
fixef(object,...)

## S3 method for class 'hlme'
ranef(object,...)
## S3 method for class 'lcmm'
ranef(object,...)
## S3 method for class 'Jointlcmm'
ranef(object,...)
## S3 method for class 'multlcmm'
ranef(object,...)

## S3 method for class 'hlme'
fitted(object,...)
## S3 method for class 'lcmm'
fitted(object,...)
## S3 method for class 'Jointlcmm'
fitted(object,...)
## S3 method for class 'multlcmm'
fitted(object,...)

## S3 method for class 'hlme'
```

```
residuals(object,...)
## S3 method for class 'lcmm'
residuals(object,...)
## S3 method for class 'Jointlcmm'
residuals(object,...)
## S3 method for class 'multlcmm'
residuals(object,...)
```

Arguments

`object`	an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`
`...`	other arguments. There are ignored in these functions.

Value

For `coef`, the vector of the estimates.

For `vcov`, the variance-covariance matrix of the estimates.

For `fixef` : - for `hlme`, `lcmm` and `multlcmm` objects, a list containing the fixed effects estimates in the class-membership model and in the longitudinal model. - for `Jointlcmm` objects, a list containing the fixed effects estimates in the class-membership model, the survival model and in the longitudinal model.

For `ranef`, a matrix (nrow=number of subjects, ncol=number of covariates with random effect) containing the individual random effects.

For `fitted`, a vector containing the subject-specific predictions extracted from `object`.

For `residuals`, a vector containing the subject-specific residuals extracted from `object`.

Author(s)

Cecile Proust-Lima, Viviane Philipps

`multlcmm` *Estimation of mutlivariate mixed-effect models and multivariate latent class mixed-effect models for multivariate longitudinal outcomes of possibly multiple types (continuous Gaussian, continuous non-Gaussian - curvilinear) that measure the same underlying latent process.*

Description

This function constitutes a multivariate extension of function `lcmm`. It fits multivariate mixed models and multivariate latent class mixed models for multivariate longitudinal outcomes of different types. It handles continuous longitudinal outcomes (Gaussian or non-Gaussian, curvilinear) as well as bounded quantitative and discrete longitudinal outcomes. Next version will also handle ordinal outcomes. The model assumes that all the outcomes measure the same underlying latent process defined as their common factor, and each outcome is related to this latent common factor by a specific parameterized link function. At the latent process level, the model estimates a standard linear mixed model or a latent class linear mixed model when heterogeneity in the population is investigated (in the same way as in function hlme). Parameters of the nonlinear link functions and of the latent process mixed model are estimated simultaneously using a maximum likelihood method.

Usage

```
multlcmm(fixed, mixture, random, subject, classmb, ng = 1,
idiag = FALSE,nwg = FALSE, randomY=FALSE, link = "linear",
intnodes = NULL, epsY = 0.5, cor=NULL, data, B, convB = 1e-04,
convL = 1e-04, convG = 1e-04, maxiter=100,
nsim=100, prior,range=NULL, subset=NULL, na.action=1,
posfix=NULL, partialH=FALSE, verbose=TRUE)
```

Arguments

- `fixed` — a two-sided linear formula object for specifying the fixed-effects in the linear mixed model at the latent process level. The response outcomes are separated by `+` on the left of `~` and the covariates are separated by `+` on the right of the `~`. For identifiability purposes, the intercept specified by default should not be removed by a `-1`. Variables on which a contrast above the different outcomes should also be estimated are included with `contrast()`.
- `mixture` — a one-sided formula object for the class-specific fixed effects in the latent process mixed model (to specify only for a number of latent classes greater than 1). Among the list of covariates included in `fixed`, the covariates with class-specific regression parameters are entered in `mixture` separated by `+`. By default, an intercept is included. If no intercept, `-1` should be the first term included.
- `random` — an optional one-sided formula for the random-effects in the latent process mixed model. At least one random effect should be included for identifiability purposes. Covariates with a random-effect are separated by `+`. By default, an intercept is included. If no intercept, `-1` should be the first term included.
- `subject` — name of the covariate representing the grouping structure.
- `classmb` — an optional one-sided formula describing the covariates in the class-membership multinomial logistic model. Covariates included are separated by `+`. No intercept should be included in this formula.
- `ng` — number of latent classes considered. If `ng=1` no `mixture` nor `classmb` should be specified. If `ng>1`, `mixture` is required.
- `idiag` — optional logical for the variance-covariance structure of the random-effects. If `FALSE`, a non structured matrix of variance-covariance is considered (by default). If `TRUE` a diagonal matrix of variance-covariance is considered.
- `nwg` — optional logical of class-specific variance-covariance of the random-effects. If `FALSE` the variance-covariance matrix is common over latent classes (by default). If `TRUE` a class-specific proportional parameter multiplies the variance-covariance matrix in each class (the proportional parameter in the last latent class equals 1 to ensure identifiability).

randomY
: optional logical for including an outcome-specific random intercept. If FALSE no outcome-specific random intercept is added (default). If TRUE independent outcome-specific random intercepts with parameterized variance are included.

link
: optional vector of families of parameterized link functions to estimate (one by outcome). Option "linear" (by default) specifies a linear link function. Other possibilities include "beta" for estimating a link function from the family of Beta cumulative distribution functions and "Splines" for approximating the link function by I-splines. For this latter case, the number of nodes and the nodes location should be also specified. The number of nodes is first entered followed by -, then the location is specified with "equi", "quant" or "manual" for respectively equidistant nodes, nodes at quantiles of the marker distribution or interior nodes entered manually in argument `intnodes`. It is followed by - and finally "splines" is indicated. For example, "7-equi-splines" means I-splines with 7 equidistant nodes, "6-quant-splines" means I-splines with 6 nodes located at the quantiles of the marker distribution and "9-manual-splines" means I-splines with 9 nodes, the vector of 7 interior nodes being entered in the argument `intnodes`.

intnodes
: optional vector of interior nodes. This argument is only required for a I-splines link function with nodes entered manually.

epsY
: optional definite positive real used to rescale the marker in (0,1) when the beta link function is used. By default, epsY=0.5.

cor
: optional indicator for inclusion of an autocorrelated Gaussian process in the latent process linear (latent process) mixed model. Option "BM" indicates a brownian motion with parameterized variance. Option "AR" specifies an autoregressive process of order 1 with parameterized variance and correlation intensity. Each option should be followed by the time variable in brackets as `cor=BM(time)`. By default, no autocorrelated Gaussian process is added.

data
: data frame containing the variables named in `fixed`, `mixture`, `random`, `classmb` and `subject`.

B
: optional specification for the initial values for the parameters. Three options are allowed: (1) a vector of initial values is entered (the order in which the parameters are included is detailed in `details` section). (2) nothing is specified. A preliminary analysis involving the estimation of a standard linear mixed model is performed to choose initial values. (3) when ng>1, a multlcmm object is entered. It should correspond to the exact same structure of model but with ng=1. The program will automatically generate initial values from this model. This specification avoids the preliminary analysis indicated in (2) Note that due to possible local maxima, the `B` vector should be specified and several different starting points should be tried.

convB
: optional threshold for the convergence criterion based on the parameter stability. By default, convB=0.0001.

convL
: optional threshold for the convergence criterion based on the log-likelihood stability. By default, convL=0.0001.

convG
: optional threshold for the convergence criterion based on the derivatives. By default, convG=0.0001.

maxiter
: optional maximum number of iterations for the Marquardt iterative algorithm. By default, maxiter=100.

nsim	number of points used to plot the estimated link functions. By default, nsim=100.
prior	name of the covariate containing the prior on the latent class membership. The covariate should be an integer with values in 0,1,...,ng. When there is no prior, the value should be 0. When there is a prior for the subject, the value should be the number of the latent class (in 1,...,ng).
range	optional vector indicating the range of the outcomes (that is the minimum and maximum). By default, the range is defined according to the minimum and maximum observed values of the outcome. The option should be used only for Beta and Splines transformations.
subset	optional vector giving the subset of observations in data to use. By default, all lines.
na.action	Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.
posfix	Optional vector giving the indices in vector B of the parameters that should not be estimated. Default to NULL, all parameters are estimated.
partialH	optional logical for Beta or Splines link functions only. Indicates whether the parameters of the link functions can be dropped from the Hessian matrix to define convergence criteria.
verbose	logical indicating if information about computation should be reported. Default to TRUE.

Details

A. THE PARAMETERIZED LINK FUNCTIONS

multlcmm function estimates multivariate latent class mixed models for different types of outcomes by assuming a parameterized link function for linking each outcome Y_k(t) with the underlying latent common factor L(t) they measure. To fix the latent process dimension, we chose to constrain at the latent process level the (first) intercept of the latent class mixed model at 0 and the standard error of the first random effect at 1.

1. With the "linear" link function, 2 parameters are required for the following transformation (Y(t) - b1)/b2

2. With the "beta" link function, 4 parameters are required for the following transformation: [h(Y(t)',b1,b2) - b3]/b4 where h is the Beta CDF with canonical parameters c1 and c2 that can be derived from b1 and b2 as c1=exp(b1)/[exp(b2)*(1+exp(b1))] and c2=1/[exp(b2)*(1+exp(b1))], and Y(t)' is the rescaled outcome i.e. Y(t)'= [Y(t) - min(Y(t)) + epsY] / [max(Y(t)) - min(Y(t)) +2*epsY].

3. With the "splines" link function, n+2 parameters are required for the following transformation b_1 + b_2*I_1(Y(t)) + ... + b_{n+2} I_{n+1}(Y(t)), where I_1,...,I_{n+1} is the basis of quadratic I-splines. To constraint the parameters to be positive, except for b_1, the program estimates b_k^* (for k=2,...,n+2) so that b_k=(b_k^*)^2. This parameterization may lead in some cases to problems of convergence that we are currently addressing.

Details of these parameterized link functions can be found in the papers: Proust-Lima et al. (Biometrics 2006) and Proust-Lima et al. (BJMSP 2013).

B. THE VECTOR OF PARAMETERS B

The parameters in the vector of initial values `B` or in the vector of maximum likelihood estimates `best` are included in the following order: (1) ng-1 parameters are required for intercepts in the latent class membership model, and if covariates are included in `classmb`, ng-1 paramaters should be entered for each one; (2) for all covariates in `fixed`, one parameter is required if the covariate is not in `mixture`, ng paramaters are required if the covariate is also in `mixture`; When ng=1, the intercept is not estimated and no parameter should be specified in `B`. When ng>1, the first intercept is not estimated and only ng-1 parameters should be specified in `B`; (3) for all covariates included with `contrast()` in `fixed`, one supplementary parameter per outcome is required excepted for the last outcome for which the parameter is not estimated but deduced from the others; (4) if `idiag=TRUE`, the variance of each random-effect specified in `random` is required excepted the first one (usually the intercept) which is constrained to 1. (5) if `idiag=FALSE`, the inferior triangular variance-covariance matrix of all the random-effects is required excepted the first variance (usually the intercept) which is constrained to 1. (5) only if `nwg=TRUE` and `ng`>1, ng-1 parameters for class-specific proportional coefficients for the variance covariance matrix of the random-effects; (6) if `cor` is specified, the standard error of the Brownian motion or the standard error and the correlation parameter of the autoregressive process; (7) the standard error of the outcome-specific Gaussian errors (one per outcome); (8) if `randomY=TRUE`, the standard error of the outcome-specific random intercept (one per outcome); (9) the parameters of each parameterized link function: 2 for "linear", 4 for "beta", n+2 for "splines" with n nodes.

C. CAUTIONS REGARDING THE USE OF THE PROGRAM

Some caution should be made when using the program. Convergence criteria are very strict as they are based on the derivatives of the log-likelihood in addition to the parameter and log-likelihood stability. In some cases, the program may not converge and reach the maximum number of iterations fixed at 100. In this case, the user should check that parameter estimates at the last iteration are not on the boundaries of the parameter space.

If the parameters are on the boundaries of the parameter space, the identifiability of the model is critical. This may happen especially with splines parameters that may be too close to 0 (lower boundary) or classmb parameters that are too high or low (perfect classification). When identifiability of some parameters is suspected, the program can be run again from the former estimates by fixing the suspected parameters to their value with option posfix. This usually solves the problem. An alternative is to remove the parameters of the Beta of Splines link function from the inverse of the Hessian with option partialH.

If not, the program should be run again with other initial values, with a higher maximum number of iterations or less strict convergence tolerances.

Specifically when investigating heterogeneity (that is with ng>1): (1) As the log-likelihood of a latent class model can have multiple maxima, a careful choice of the initial values is crucial for ensuring convergence toward the global maximum. The program can be run without entering the vector of initial values (see point 2). However, we recommend to systematically enter initial values in `B` and try different sets of initial values. (2) The automatic choice of initial values we provide requires the estimation of a preliminary linear mixed model. The user should be aware that first, this preliminary analysis can take time for large datatsets and second, that the generated initial values can be very not likely and even may converge slowly to a local maximum. This is the reason why several alternatives exist. The vector of initial values can be directly specified in `B` the initial values can be generated (automatically or randomly) from a model with `ng=`. Finally, function `gridsearch` performs an automatic grid search.

Value

The list returned is:

- `ns` number of grouping units in the dataset
- `ng` number of latent classes
- `loglik` log-likelihood of the model
- `best` vector of parameter estimates in the same order as specified in `B` and detailed in section `details`
- `V` vector containing the upper triangle matrix of variance-covariance estimates of `Best` with exception for variance-covariance parameters of the random-effects for which `V` contains the variance-covariance estimates of the Cholesky transformed parameters displayed in `cholesky`
- `gconv` vector of convergence criteria: 1. on the parameters, 2. on the likelihood, 3. on the derivatives
- `conv` status of convergence: =1 if the convergence criteria were satisfied, =2 if the maximum number of iterations was reached, =4 or 5 if a problem occured during optimisation
- `call` the matched call
- `niter` number of Marquardt iterations
- `N` internal information used in related functions
- `idiag` internal information used in related functions
- `pred` table of individual predictions and residuals in the underlying latent process scale; it includes marginal predictions (pred_m), marginal residuals (resid_m), subject-specific predictions (pred_ss) and subject-specific residuals (resid_ss) averaged over classes, the transformed observations in the latent process scale (obs) and finally the class-specific marginal and subject-specific predictions (with the number of the latent class: pred_m_1,pred_m_2,...,pred_ss_1,pred_ss_2,...).
- `pprob` table of posterior classification and posterior individual class-membership probabilities
- `Xnames` list of covariates included in the model
- `predRE` table containing individual predictions of the random-effects : a column per random-effect, a line per subject.
- `cholesky` vector containing the estimates of the Cholesky transformed parameters of the variance-covariance matrix of the random-effects
- `estimlink` table containing the simulated values of each outcome and the corresponding estimated link function
- `epsY` definite positive reals used to rescale the markers in (0,1) when the beta link function is used. By default, epsY=0.5.
- `linktype` indicators of link function types: 0 for linear, 1 for beta, 2 for splines and 3 for thresholds
- `linknodes` vector of nodes useful only for the 'splines' link functions

Author(s)

Cecile Proust-Lima and Viviane Philipps

<cecile.proust-lima@inserm.fr>

References

Proust-Lima C, Philipps V, Liquet B (2017). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: the R package lcmm. Journal of Statistical Software, 78(2), 1-56

Proust and Jacqmin-Gadda (2005). Estimation of linear mixed models with a mixture of distribution for the random-effects. Comput Methods Programs Biomed 78: 165-73.

Proust, Jacqmin-Gadda, Taylor, Ganiayre, and Commenges (2006). A nonlinear model with latent process for cognitive evolution using multivariate longitudinal data. Biometrics 62, 1014-24.

Proust-Lima, Dartigues and Jacqmin-Gadda (2011). Misuse of the linear mixed model when evaluating risk factors of cognitive decline. Amer J Epidemiol 174(9): 1077-88.

Proust-Lima, Amieva, Jacqmin-Gadda (2013). Analysis of multivariate mixed longitudinal data: A flexible latent process approach. Br J Math Stat Psychol 66(3): 470-87.

Commenges, Proust-Lima, Samieri, Liquet (2015). A universal approximate cross-validation criterion for regular risk functions. Int J Biostat. 2015 May; 11(1):51-67

See Also

postprob, plot.multlcmm, predictL, predictY lcmm

Examples

```
## Not run:
# Latent process mixed model for two curvilinear outcomes. Link functions are
# aproximated by I-splines, the first one has 3 nodes (i.e. 1 internal node 8),
# the second one has 4 nodes (i.e. 2 internal nodes 12,25)

m1 <- multlcmm(Ydep1+Ydep2~1+Time*X2+contrast(X2),random=~1+Time,
subject="ID",randomY=TRUE,link=c("4-manual-splines","3-manual-splines"),
intnodes=c(8,12,25),data=data_lcmm)

# to reduce the computation time, the same model is estimated using
# a vector of initial values
m1 <- multlcmm(Ydep1+Ydep2~1+Time*X2+contrast(X2),random=~1+Time,
subject="ID",randomY=TRUE,link=c("4-manual-splines","3-manual-splines"),
intnodes=c(8,12,25),data=data_lcmm,
B=c(-1.071, -0.192,  0.106, -0.005, -0.193,  1.012,  0.870,  0.881,
  0.000,  0.000, -7.520,  1.401,  1.607 , 1.908,  1.431,  1.082,
 -7.528,  1.135 , 1.454 , 2.328, 1.052))


# output of the model
summary(m1)
# estimated link functions
plot(m1,which="linkfunction")
# variation percentages explained by linear mixed regression
```

```
VarExpl(m1,data.frame(Time=0))

#### Heterogeneous latent process mixed model with linear link functions
#### and 2 latent classes of trajectory
m2 <- multlcmm(Ydep1+Ydep2~1+Time*X2,random=~1+Time,subject="ID",
link="linear",ng=2,mixture=~1+Time,classmb=~1+X1,data=data_lcmm,
B=c( 18,-20.77,1.16,-1.41,-1.39,-0.32,0.16,-0.26,1.69,1.12,1.1,10.8,
1.24,24.88,1.89))
# summary of the estimation
summary(m2)
# posterior classification
postprob(m2)
# longitudinal predictions in the outcomes scales for a given profile of covariates
newdata <- data.frame(Time=seq(0,5,length=100),X1=rep(0,100),X2=rep(0,100),X3=rep(0,100))
predGH <- predictY(m2,newdata,var.time="Time",methInteg=0,nsim=20)
head(predGH)

## End(Not run)
```

---

paquid *Longitudinal data on cognitive and physical aging in the elderly*

Description

The dataset consists in a subsample of the Paquid prospective cohort study. Repeated measures cognitive measures (MMSE, IST, BVRT psychometric tests), physical dependency (HIER) and depression sympatomatology (CESD) were collected over a maximum period of 20 years along with dementia information (age at dementia diagnosis, dementia diagnosis information). Time-independent socio-demographic information is also provided (CEP, male, age_init).

Usage

```
paquid
```

Format

A data frame with 2250 observations over 500 subjects and 12 variables:

- `ID` subject identification number
- `MMSE` score at the Mini-Mental State Examination (MMSE), a psychometric test of global cognitive functioning (integer in range 0-30)
- `BVRT` score at the Benton Visual Retention Test (BVRT), a psychometric test of spatial memory (integer in range 0-15)
- `IST` score at the Isaacs Set Test (IST) truncated at 15 seconds, a test of verbal memory (integer in range 0-40)
- `HIER` score of physical dependency (0=no dependency, 1=mild dependency, 2=moderate dependency, 3=severe dependency)

CESD score of a short self-report scale CES-D designed to measure depressive symptomatology in the general population (integer in range 0-52)

age age at the follow-up visit

dem indicator of positive diagnosis of dementia

agedem age at dementia diagnosis for dem=1 and at last contact for dem=0

age_init age at entry in the cohort

CEP binary indicator of educational level (CEP=1 for subjects who graduated from primary school; CEP=0 otherwise)

male binary indicator for gender (male=1 for men; male=0 for women)

References

Letenneur, L., Commenges, D., Dartigues, J. F., & Barberger-Gateau, P. (1994). Incidence of dementia and Alzheimer's disease in elderly community residents of southwestern France. International Journal of Epidemiology, 23 (6), 1256-61.

Examples

```
summary(paquid)
```

---

plot.cuminc *Plot of predicted cumulative incidences according to a profile of covariates*

Description

This function displays the predicted cause-specific cumulative incidences derived from a joint latent class model according to a profile of covariates.

Usage

```
## S3 method for class 'cuminc'
plot(x, profil, event, add = FALSE, legend, legend.loc = "topleft",  ...)
```

Arguments

x	an object of class cuminc
profil	an integer giving the profile number for which the cumulative incidences are to be plotted.
event	an integer giving the event indicator for which the cumulative incidence are to be plotted.
add	logical indicating if the curves should be added to an existing plot. Default to FALSE.
legend	character or expression to appear in the legend. If no legend should be added, "legend" should be NULL.

`legend.loc`	keyword for the position of the legend from the list `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"topleft"`,`"top"`, `"topright"`, `"right"` and `"center"`. By default, the legend is located in the top left of the plot.
`...`	other parameters to be passed through to plotting functions

Value

returns NULL

Author(s)

Viviane Philipps and Cecile Proust-Lima

See Also

`Jointlcmm`,`plot.Jointlcmm`,`cuminc`

`plot.dynpred` *Plot of individual dynamic predictions*

Description

This function provides a graphical representation of individual dynamic predictions obtained from a joint latent class model and plots simultaneously the observed outcome.

Usage

```
## S3 method for class 'dynpred'
plot(x,subject=NULL,landmark=NULL,horizon=NULL,
add=FALSE,...)
```

Arguments

`x`	a dynpred object, containing the predicted probabilities of event in a time window, obtained from a joint latent class model.
`subject`	a vector containing the identifiers of the subjects the user wants to display. If NULL (the default), all subjects are plotted.
`landmark`	a vector containing the landmark times from which the probabilities are to be plotted. If NULL (the default), all landmarks are used. If several horizon are specified, only one landmark should be selected.
`horizon`	a vector containing the horizon times from which the probabilities are to be plotted. If NULL (the default), all horizons are used. If several landmarks are specified, only one horizon should be selected.
`add`	logical indicating if the plot should be added to an existing plot. By default (add=FALSE), a new plot is created.
`...`	optional graphical parameters.

Details

Two types of plot are provided for the moment :

- if one horizon is selected (and one or several landmarks), each prediction is represented by a point at the landmark time. If available, the predictions are surrounded by confidence intervals.

- if several horizons (t1, t2, etc) and only one landmark (s) is selected, a line linking the predictions (placed at abscissa s+t1, s+t2, etc) is drawn. Confidence bands (if available) are represented as dotted lines.

Value

returns NULL

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

dynpred

Examples

```
## Not run:

## Joint latent class model with 2 classes :
m32 <- Jointlcmm(Ydep1~Time*X1,mixture=~Time,random=~Time,subject="ID",
classmb=~X3,ng=2,survival=Surv(Tevent,Event)~X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",data=data_lcmm,B = c(0.64, -0.62,
0, 0, 0.52, 0.81, 0.41, 0.78, 0.1, 0.77, -0.05, 10.43, 11.3, -2.6, -0.52, 1.41,
-0.05, 0.91, 0.05, 0.21, 1.5))

## Predictions at landmark 10 and 12 for horizon 3, 5 and 10 for two subjects :
dynpred.m32 <- dynpred(m32,landmark=c(10,12),horizon=c(3,5,10),var.time="Time",
fun.time=function(x){10*x},newdata=data_lcmm[4:8,],draws=TRUE,ndraws=2000)

## Plot of the predictions at landmark 10 for horizon 3,5,10 :
plot(dynpred.m32,landmark=10)

## Plot of the predictions at landmark 10 and 12 for horizon 3 :
plot(dynpred.m32,horizon=3)

## End(Not run)
```

plot.lcmm *Plot of a fitted model*

Description

This function produces different plots (residuals, goodness-of-fit, estimated link functions, estimated baseline risk/survival and posterior probabilities distributions) of a fitted object of class hlme, lcmm, multlcmm or Jointlcmm.

Usage

```
## S3 method for class 'hlme'
plot(x,which="residuals",var.time,break.times,marg,subset,...)
## S3 method for class 'lcmm'
plot(x,which="residuals",var.time,break.times,marg,subset,...)
## S3 method for class 'Jointlcmm'
plot(x,which="residuals",var.time,break.times,marg,event,subset,...)
## S3 method for class 'multlcmm'
plot(x,which="residuals",var.time,break.times,marg,outcome,subset,...)
```

Arguments

`x`	an object inheriting from classes `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`, representing respectively a fitted latent class linear mixed model, a more general latent class mixed model or a joint latent class model
`which`	a character string indicating the type of plot to produce. For `hlme` objects, are available "residuals", "postprob","fit". For `lcmm` and `multlcmm` objects, are available "residuals", "postprob", "link", "linkfunction", "fit". For `Jointlcmm` objects, are avaiable "residuals", "postprob", "link", "linkfunction", "fit", "hazard", "baselinerisk", "survival". Default to "residuals"
`var.time`	for `which="fit"` only, a character string containing the name of the variable that corresponds to time in the longitudinal model.
`break.times`	for `which="fit"` only, either a numeric vector containing the cuts-off defining the time-intervals or an integer giving the number of cut-offs. In the latter case, the cut-offs are placed at the quantiles of the observed times distribution.
`marg`	for `which="fit"` only, a logical indicating the type of prediction. If `marg=TRUE` (the default), the marginal predictions are provided. If `marg=FALSE`, the subject-specific predictions are provided.
`outcome`	for `which="fit"` and `multlcmm` objects only, the outcome to consider.
`event`	for `which="baselinerisk"` or `which="hazard"` only, an integer corresponding to the numeric code (in the indicator variable) of the event for which the baseline risk functions are to be plotted. By default, the first event is considered.
`subset`	for `which="fit"` only, a subset of the data used to estimate the model, defining the data on which the fit is evaluated. By default, all the data are used.
`...`	other parameters to be passed through to plotting functions

Details

With which="residuals", this function provides the marginal residuals against the marginal predictions, the subject-specific residuals against the subject-specific predictions, a normal QQ-plot with confidence bands for the marginal residuals and a normal QQ-plot with confidence bands for the subject-specific residuals.

With which="postprob", the function provides the histograms of the posterior class-membership probabilities stemmed from a Jointlcmm, lcmm, hlme or multlcmm object.

With which="link" or which="linkfunction", the function displays the estimated transformation(s) specified in the option link of lcmm and multlcmm functions. It corresponds to the (non)linear parameterized link estimated between the oberved longitudinal outcome and the underlying latent process.

With which="fit", the function provides the class-specific weighted marginal and subject-specific mean predicted trajectories with time and the class-specific weighted mean observed trajectories and their 95% confidence bounds. The predicted and observed class-specific values are weighted means within each time interval; For each observation or prediction (in the transformed scale if appropriate), the weights are the class-specific (posterior with subject-specific or marginal otherwise) probabilities to belong to the latent class.

With which="baselinerisk" or which="hazard", the function displays the estimated baseline risk functions for the time-to-event of interest in each latent class.

With which="survival", the function displays the estimated event-free probabilities (survival functions) for the time-to-event of interest in each latent class.

Author(s)

Cecile Proust-Lima, Viviane Philipps and Benoit Liquet

See Also

hlme, lcmm, multlcmm, Jointlcmm

Examples

```
###################### fit, residuals and postprob

# estimation of the model
m<-lcmm(Y~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,
subject='ID',ng=2,data=data_hlme,B=c(0.41,0.55,-0.18,-0.41,
-14.26,-0.34,1.33,13.51,24.65,2.98,1.18,26.26,0.97))

# fit
plot(m,which="fit",marg=FALSE,var.time="Time",bty="n")
# residuals plot
plot(m)
# postprob plot
plot(m,which="postprob")
```

```
##################### fit, linkfunctions

#### Estimation of homogeneous mixed models with different assumed link
#### functions, a quadratic mean trajectory for the latent process with
#### independent random intercept, slope and quadratic slope
#### (comparison of linear, Beta and 3 and 5 splines link functions)
## Not run:

# linear link function
m10<-lcmm(Ydep2~Time+I(Time^2),random=~Time+I(Time^2),subject='ID',ng=1,
          data=data_lcmm,link="linear",
          B=c(-0.7454, -0.2031,  0.2715,  0.2916 , 0.6114, -0.0064,  0.0545,
              0.0128, 25.3795, 2.2371))

# Beta link function
m11<-lcmm(Ydep2~Time+I(Time^2),random=~Time+I(Time^2),subject='ID',ng=1,
          data=data_lcmm,link="beta",B=c(-0.9109, -0.0831,  0.5194,  0.1910 ,
          0.8984, -0.0179, -0.0636,  0.0045,  0.5514, -0.7692,  0.7037,  0.0899))

# fit
par(mfrow=c(2,1),mar=c(4,4,1,1))
plot(m11,which="fit",var.time="Time",bty="l",ylim=c(-3,0))
plot(m11,which="fit",var.time="Time",marg=FALSE,bty="l",ylim=c(-3,0))

# I-splines with 3 equidistant nodes
m12<-lcmm(Ydep2~Time+I(Time^2),random=~Time+I(Time^2),subject='ID',ng=1,
          data=data_lcmm,link="3-equi-splines",B=c(-0.9272, -0.0753 , 0.5304,
          0.1950,  0.9260, -0.0204, -0.0739 , 0.0059, -7.8369,  0.9228 ,-1.4689,
          2.0396,  1.8102))

# I-splines with 5 nodes, and interior nodes entered manually
m13<-lcmm(Ydep2~Time+I(Time^2),random=~Time+I(Time^2),subject='ID',ng=1,
          data=data_lcmm,link="5-manual-splines",intnodes=c(10,20,25),
          B=c(-0.9315, -0.0739 , 0.5254 , 0.1933,  0.9418, -0.0206, -0.0776,
          0.0064, -7.8645, 0.7470,  1.2080,  1.5537 , 1.7558 , 1.3386 , 1.0982))

# Plot of estimated different link functions:
# (applicable for models that only differ in the "link function" used.
# Otherwise, the latent process scale is different and a rescaling
# is necessary)
plot(m10,which="linkfunction",bty="l")
plot(m11,which="linkfunction",bty="l",add=TRUE,col=2)
plot(m12,which="linkfunction",bty="l",add=TRUE,col=3)
plot(m13,which="linkfunction",bty="l",add=TRUE,col=4)
legend("topleft",col=1:4,legend=c("linear","beta","3-Isplines","5-Isplines"),lty=1,bty='n')

## End(Not run)


##################### fit, baselinerisk and survival
## Not run:
#### estimation with 3 latent classes (ng=3) - see Jointlcmm
#### help for details on the model
```

```
m3 <- Jointlcmm(fixed= Ydep1~Time*X1,mixture=~Time,random=~Time,
classmb=~X3,subject='ID',survival = Surv(Tevent,Event)~ X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",ng=3,data=data_lcmm,
B=c(0.7576, 0.4095, -0.8232, -0.2737, 0, 0, 0, 0.2838, -0.6338,
2.6324, 5.3963, -0.0273, 1.3979, 0.8168, -15.041, 10.164, 10.2394,
11.5109, -2.6219, -0.4553, -0.6055, 1.473, -0.0383, 0.8512, 0.0389,
0.2624, 1.4982))

# fit
plot(m3,which="fit",var.time="Time",bty="l")
plot(m3,which="fit",var.time="Time",marg=FALSE,bty="l",ylim=c(0,15))


# Class-specific predicted baseline risk & survival functions in the
# 3-class model retained (for the reference value of the covariates)
plot(m3,which="baselinerisk",bty="l")
plot(m3,which="baselinerisk",ylim=c(0,5),bty="l")
plot(m3,which="survival",bty="l")

## End(Not run)
```

plot.pred.accuracy *Plots*

Description

This function displays plots related to predictive accuracy functions: epoce and Diffepoce.

These functions do not apply for the moment with multiple causes of event (competing risks).

Usage

```
## S3 method for class 'epoce'
plot(x,...)
## S3 method for class 'Diffepoce'
plot(x,...)
```

Arguments

x	an object inheriting from classes epoce or Diffepoce
...	other parameters to be passed through to plotting functions

Details

For epoce objects, the function displays the EPOCE estimate (either MPOL or CVPOL) according to the time of prediction. For Diffepoce objects, plot displays the difference in EPOCE estimates (either MPOL or CVPOL) and its 95% tracking interval between two joint latent class models

Value

Returns plots related to epoce and Diffepoce

Author(s)

Cecile Proust-Lima and Viviane Philipps

See Also

epoce,Diffepoce

Examples

```
## Not run:
# estimation of the joint latent class model
m3 <- Jointlcmm(fixed= Ydep1~Time*X1,mixture=~Time,random=~Time,
classmb=~X3,subject='ID',survival = Surv(Tevent,Event)~X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",ng=3,data=data_lcmm,
B=c(0.7667, 0.4020, -0.8243, -0.2726, 0.0000, 0.0000, 0.0000, 0.3020,
-0.6212, 2.6247, 5.3139, -0.0255, 1.3595, 0.8172, -11.6867, 10.1668,
10.2355, 11.5137, -2.6209, -0.4328, -0.6062, 1.4718, -0.0378, 0.8505,
0.0366, 0.2634, 1.4981))
# predictive accuracy of the model evaluated with EPOCE
VecTime <- c(1,3,5,7,9,11,13,15)
cvpl <- epoce(m3,var.time="Time",pred.times=VecTime)
summary(cvpl)
plot(cvpl,bty="l",ylim=c(0,2))

## End(Not run)
```

---

plot.predict *Plot of predicted trajectories and link functions*

Description

This function provides the class-specific predicted trajectories stemmed from a hlme, lcmm, multlcmm or Jointlcmm object.

Usage

```
## S3 method for class 'predictL'
plot(x,legend.loc="topright",legend,add=FALSE,shades=FALSE,...)
## S3 method for class 'predictY'
plot(x,outcome=1,legend.loc="topright",legend,add=FALSE,shades=FALSE,...)
## S3 method for class 'predictlink'
plot(x,legend.loc="topleft",legend,add=FALSE,shades=FALSE,...)
```

Arguments

x	an object inheriting from classes `predictL`, `predictY` or `predictlink` representing respectively the predicted marginal mean trajectory of the latent process, the predicted marginal mean trajectory of the longitudinal outcome, or the predicted link function of a fitted latent class model.
outcome	for `predictY` and multivariate model fitted with `multlcmm` only, the outcome to consider.
legend.loc	keyword for the position of the legend from the list `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"topleft"`,`"top"`, `"topright"`, `"right"` and `"center"`.
legend	character or expression to appear in the legend. If no legend should be added, `"legend"` should be NULL.
add	logical indicating if the curves should be added to an existing plot. Default to FALSE.
shades	logical indicating if confidence intervals should be represented with shades. Default to FALSE, the confidence intervals are represented with dotted lines.
...	other parameters to be passed through to plotting functions or to legend

Author(s)

Cecile Proust-Lima, Benoit Liquet and Viviane Philipps

See Also

`hlme`, `lcmm`, `Jointlcmm`, `multlcmm`

Examples

```
################# Prediction from linear latent class model
## fitted model
m<-lcmm(Y~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,
subject='ID',ng=2,data=data_hlme,B=c(0.41,0.55,-0.18,-0.41,
-14.26,-0.34,1.33,13.51,24.65,2.98,1.18,26.26,0.97))
## newdata for predictions plot
newdata<-data.frame(Time=seq(0,5,length=100),
X1=rep(0,100),X2=rep(0,100),X3=rep(0,100))
plot(predictL(m,newdata,var.time="Time"),legend.loc="right",bty="l")
## data from the first subject for predictions plot
firstdata<-data_hlme[1:3,]
plot(predictL(m,firstdata,var.time="Time"),legend.loc="right",bty="l")

 ## Not run:
################# Prediction from a joint latent class model
## fitted model - see help of Jointlcmm function for details on the model
m3 <- Jointlcmm(fixed= Ydep1~Time*X1,mixture=~Time,random=~Time,
classmb=~X3,subject='ID',survival = Surv(Tevent,Event)~X1+mixture(X2),
hazard="3-quant-splines",hazardtype="PH",ng=3,data=data_lcmm,
B=c(0.7576, 0.4095, -0.8232, -0.2737, 0, 0, 0, 0.2838, -0.6338,
2.6324, 5.3963, -0.0273, 1.398, 0.8168, -15.041, 10.164, 10.2394,
```

```
11.5109, -2.6219, -0.4553, -0.6055, 1.473, -0.0383, 0.8512, 0.0389,
0.2624, 1.4982))
# class-specific predicted trajectories
#(with characteristics of subject ID=193)
data <- data_lcmm[data_lcmm$ID==193,]
plot(predictY(m3,newdata=data,var.time="Time"),bty="l")

## End(Not run)
```

---

postprob — *Posterior classification stemmed from a* `hlme`*,* `lcmm`*,* `multlcmm` *or* `Jointlcmm` *estimation*

Description

This function provides informations about the posterior classification stemmed from a `hlme`, `lcmm`, `multlcmm` or `Jointlcmm` object.

Usage

```
## S3 method for class 'hlme'
postprob(x,threshold=c(0.7,0.8,0.9),...)
## S3 method for class 'lcmm'
postprob(x,threshold=c(0.7,0.8,0.9),...)
## S3 method for class 'Jointlcmm'
postprob(x,threshold=c(0.7,0.8,0.9),...)
## S3 method for class 'multlcmm'
postprob(x,threshold=c(0.7,0.8,0.9),...)
```

Arguments

`x`	an object inheriting from classes `hlme`, `lcmm`, `Jointlcmm` or `multlcmm` representing respectively a fitted latent class linear mixed-effects model, a more general latent class mixed model, a joint latent class model or a multivariate general latent class mixed model.
`threshold`	optional vector of thresholds for the posterior probabilities
`...`	further arguments to be passed to or from other methods. They are ignored in this function.

Details

This function provides the number of subjects classified a posteriori in each latent class, the percentage of subjects classified with a posterior probability above a certain threshold, and the classification table that contains the mean of the posterior probability of belonging to each latent class over the subjects classified in each of the latent classes. This table aims at evaluating the quality of the posterior classification. For `hlme`, `lcmm` objects, the posterior classification and the classification table are derived from the posterior class-membership probabilities given the vector of repeated measures that are contained in pprob output matrix. For a `Jointlcmm` object, the first posterior classification and the classification table are derived from the posterior class-membership probabilities given the vector of repeated measures and the time-to-event information (that are contained in columns probYT1, probYT2, etc in pprob output matrix). The second posterior classification is derived from the posterior class-membership probabilities given only the vector of repeated measures (that are contained in columns probY1, probY2, etc in pprob output matrix).

Value

A list containing the posterior classification, the posterior classification table and the percentage of subjects classified with a posterior probability above the given thresholds.

Note

This function can only be used with latent class mixed models and joint latent class mixed models that include at least 2 latent classes

Author(s)

Cecile Proust-Lima, Benoit Liquet and Viviane Philipps

See Also

`Jointlcmm`, `lcmm`, `hlme`,`plot.lcmm`

Examples

```
m<-lcmm(Y~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,
subject='ID',ng=2,data=data_hlme,B=c(0.41,0.55,-0.18,-0.41,
-14.26,-0.34,1.33,13.51,24.65,2.98,1.18,26.26,0.97))
postprob(m)
```

`predictL` *Class-specific marginal predictions in the latent process scale for* `lcmm`*,* `Jointlcmm` *and* `multlcmm` *objects*

Description

This function provides a matrix containing the class-specific predicted trajectories computed in the latent process scale, that is the latent process underlying the curvilinear outcome(s), for a profile of covariates specified by the user. This function applies only to `lcmm` and `multlcmm` objects. The function `plot.predict` provides directly the plot of these class-specific predicted trajectories. The function `predictY` provides the class-specific predicted trajectories computed in the natural scale of the outcome(s).

Usage

```
## S3 method for class 'lcmm'
predictL(x,newdata,var.time,na.action=1,confint=FALSE,...)
## S3 method for class 'multlcmm'
predictL(x,newdata,var.time,na.action=1,confint=FALSE,...)
## S3 method for class 'Jointlcmm'
predictL(x,newdata,var.time,na.action=1,confint=FALSE,...)
```

Arguments

x	an object inheriting from class lcmm,multlcmm or Jointlcmm representing a (joint) (latent class) mixed model involving a latent process and estimated link function(s).
newdata	data frame containing the data from which predictions are computed. The data frame should include at least all the covariates listed in x$Xnames2. Names in the data frame should be exactly x$Xnames2 that are the names of covariates specified in lcmm or multlcmm calls.
var.time	A character string containing the name of the variable that corresponds to time in the data frame (x axis in the plot).
na.action	Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.
confint	logical indicating if confidence should be provided. Default to FALSE.
...	further arguments to be passed to or from other methods. They are ignored in this function.

Value

An object of class predictL with values :

- pred : a matrix containing the class-specific predicted values in the latent process scale, the lower and the upper limits of the confidence intervals (if calculated).

- times : the var.time variable from newdata

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

plot.predict, predictY, lcmm

Examples

```
#### Prediction from a 2-class model with a Splines link function
## Not run:
## fitted model
m<-lcmm(Ydep2~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,
```

```
subject='ID',ng=2,data=data_lcmm,link="splines",B=c(
-0.175,      -0.191,       0.654,      -0.443,
-0.345,      -1.780,       0.913,       0.016,
 0.389,       0.028,       0.083,      -7.349,
 0.722,       0.770,       1.376,       1.653,
 1.640,       1.285))
summary(m)
## predictions for times from 0 to 5 for X1=0
newdata<-data.frame(Time=seq(0,5,length=100),
X1=rep(0,100),X2=rep(0,100),X3=rep(0,100))
predictL(m,newdata,var.time="Time")
## predictions for times from 0 to 5 for X1=1
newdata$X1 <- 1
predictY(m,newdata,var.time="Time")

## End(Not run)
```

`predictlink` *Confidence intervals for the estimated link functions from* `lcmm`, `Jointlcmm` *and* `multlcmm`

Description

This function provides 95% confidence intervals around the estimated transformation given in estimlink attribute of `lcmm`, `Jointlcmm` and `multlcmm` objects. It can also be used to evaluate the link functions at other values than those given in attribute `estimlink` of `lcmm`, `Jointlcmm` or `multlcmm` object.

Usage

```
## S3 method for class 'lcmm'
predictlink(x,ndraws=2000,Yvalues,...)
## S3 method for class 'multlcmm'
predictlink(x,ndraws=2000,Yvalues,...)
## S3 method for class 'Jointlcmm'
predictlink(x,ndraws=2000,Yvalues,...)
```

Arguments

`x`	an object inheriting from classes `lcmm`, `Jointlcmm` or `multlcmm`.
`ndraws`	the number of draws that should be generated to approximate the posterior distribution of the transformed values. By default, ndraws=2000.
`Yvalues`	a vector (for a `lcmm` or `Jointlcmm` object) or a matrix (for a `multlcmm` object) containing the values at which to compute the transformation(s). Default to the values in `x$estimlink`.
`...`	other parameters (ignored)

Value

An object of class `predictlink` with values :

- `pred` :

For a `lcmm` or `Jointlcmm` object, a data frame containing the values at which the transformation is evaluated, the transformed values and the lower and the upper limits of the confidence intervals (if ndraws>0).

For a `multlcmm` object, a data frame containing the indicator of the outcome, the values at which the transformations are evaluated,the transformed values and the lower and the upper limits of the confidence intervals (if ndraws>0).

- `object` : the object from which the link function is predicted

Author(s)

Cecile Proust-Lima and Viviane Philipps

See Also

`lcmm`, `multlcmm`, `plot.lcmm`, `plot.predictlink`

Examples

```
 ## Not run:

## Univariate mixed model with splines link funciton
m14<-lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="5-manual-splines",intnodes=c(10,20,25),
B=c(-0.89255, -0.09715, 0.56335, 0.21967, 0.61937, -7.90261, 0.75149,
-1.22357, 1.55832, 1.75324, 1.33834, 1.0968))

##Transformed values of several scores and their confidence intervals
transf.m14 <- predictlink(m14,ndraws=2000,Yvalues=c(0,1,7:30))
plot(transf.m14)


## Multivariate mixed model with splines link functions
m1 <- multlcmm(Ydep1+Ydep2~1+Time*X2+contrast(X2),random=~1+Time,
subject="ID",randomY=TRUE,link=c("4-manual-splines","3-manual-splines"),
intnodes=c(8,12,25),data=data_lcmm,
B=c(-1.071, -0.192,  0.106, -0.005, -0.193,  1.012,  0.870,  0.881,
  0.000,  0.000, -7.520,  1.401,  1.607 , 1.908,  1.431,  1.082,
 -7.528,  1.135 , 1.454 , 2.328, 1.052))
##Confidence intervals for the transformed values (given in m1$estimlink)
transf.m1 <- predictlink(m1,ndraws=200)
plot(transf.m1)

## End(Not run)
```

predictY — *Marginal predictions (possibly class-specific) of a* hlme*,* lcmm*,* multlcmm *or* Jointlcmm *object in the natural scale of the longitudinal outcome(s) for a specified profile of covariates.*

Description

For hlme and Jointlcmm objects, the function computes the predicted values of the longitudinal marker in each latent class for a specified profile of covariates. For lcmm and multlcmm objects, the function computes predicted values in the natural scale of the outcomes for a specified profile of covariates. For linear and threshold links, the predicted values are computed analytically. For splines and Beta links, a Gauss-Hermite or Monte-Carlo integration are used to numerically compute the predictions. In addition, for any type of link function, confidence bands (and median) can be computed by a Monte Carlo approximation of the posterior distribution of the predicted values.

Usage

```
## S3 method for class 'lcmm'
predictY(x,newdata,var.time,methInteg=0,nsim=20,draws=FALSE,
ndraws=2000,na.action=1,...)
## S3 method for class 'hlme'
predictY(x,newdata,var.time,draws=FALSE,na.action=1,...)
## S3 method for class 'Jointlcmm'
predictY(x,newdata,var.time,methInteg=0,nsim=20,draws=FALSE,
ndraws=2000,na.action=1,...)
## S3 method for class 'multlcmm'
predictY(x,newdata,var.time,methInteg=0,nsim=20,draws=FALSE,
ndraws=2000,na.action=1,...)
```

Arguments

x	an object inheriting from class lcmm, hlme, Jointlcmm or multlcmm representing a general latent class mixed model.
newdata	data frame containing the data from which predictions are computed. The data frame should include at least all the covariates listed in x$Xnames2. Names in the data frame should be exactly x$Xnames2 that are the names of covariates specified in lcmm, hlme, Jointlcmm or multlcmm calls.
var.time	A character string containing the name of the variable that corresponds to time in the data frame (x axis in the plot).
methInteg	optional integer specifying the type of numerical integration required only for predictions with splines or Beta link functions. Value 0 (by default) specifies a Gauss-Hermite integration which is very rapid but neglects the correlation between the predicted values (in presence of random-effects). Value 1 refers to a Monte-Carlo integration which is slower but correctly account for the correlation between the predicted values.

nsim
: For a lcmm, multlcmm or Jointlcmm object only; optional number of points used in the numerical integration with splines or Beta link functions. For methInteg=0, nsim should be chosen among the following values: 5, 7, 9, 15, 20, 30, 40 or 50 (nsim=20 by default). If methInteg=1, nsim should be relatively important (more than 200).

draws
: optional boolean specifying whether median and confidence bands of the predicted values should be computed (TRUE) - whatever the type of link function. For a lcmm, multlcmm or Jointlcmm object, a Monte Carlo approximation of the posterior distribution of the predicted values is computed and the median, 2.5% and 97.5% percentiles are given. Otherwise, the predicted values are computed at the point estimate. By default, draws=FALSE.

ndraws
: For a lcmm, multlcmm or Jointlcmm object only; if draws=TRUE, ndraws specifies the number of draws that should be generated to approximate the posterior distribution of the predicted values. By default, ndraws=2000.

na.action
: Integer indicating how NAs are managed. The default is 1 for 'na.omit'. The alternative is 2 for 'na.fail'. Other options such as 'na.pass' or 'na.exclude' are not implemented in the current version.

...
: further arguments to be passed to or from other methods. They are ignored in this function.

Value

An object of class predictY with values :

- pred : a matrix with the same rows (number and order) as in newdata.

For hlme objects and lcmm or Jointlcmm with draws=FALSE, returns a matrix with ng columns corresponding to the ng class-specific vectors of predicted values computed at the point estimate

For objects of class lcmm or Jointlcmm with draws=TRUE, returns a matrix with ng*3 columns representing the ng class-specific 50%, 2.5% and 97.5% percentiles of the approximated posterior distribution of the class-specific predicted values.

For objects of class multlcmm with draws=FALSE, returns a matrix with ng+1 columns: the first column indicates the name of the outcome which is predicted and the ng subsequent columns correspond to the ng class-specific vectors of predicted values computed at the point estimate

For objects of class multlcmm with draws=TRUE, returns a matrix with ng*3+1 columns: the first column indicates the name of the outcome which is predicted and the ng*3 subsequent columns correspond to the ng class-specific 50%, 2.5% and 97.5% percentiles of the approximated posterior distribution of the class-specific predicted values.

- times : the var.time variable from newdata

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

lcmm, multlcmm, hlme, Jointlcmm

Examples

```
#### Prediction from a 2-class model with a Splines link function
## Not run:
## fitted model
m<-lcmm(Ydep2~Time*X1,mixture=~Time,random=~Time,classmb=~X2+X3,
subject='ID',ng=2,data=data_lcmm,link="splines",B=c(
-0.175,      -0.191,       0.654,      -0.443,
-0.345,      -1.780,       0.913,       0.016,
 0.389,       0.028,       0.083,      -7.349,
 0.722,       0.770,       1.376,       1.653,
 1.640,       1.285))
summary(m)
## predictions for times from 0 to 5 for X1=0
newdata<-data.frame(Time=seq(0,5,length=100),
X1=rep(0,100),X2=rep(0,100),X3=rep(0,100))
pred0 <- predictY(m,newdata,var.time="Time")
head(pred0)
## Option draws=TRUE to compute a MonteCarlo
# approximation of the predicted value distribution
# (quite long with ndraws=2000 by default)
\dontrun{
pred0MC <- predictY(m,newdata,draws=TRUE,var.time="Time")
}
## predictions for times from 0 to 5 for X1=1
newdata$X1 <- 1
pred1 <- predictY(m,newdata,var.time="Time")
## Option draws=TRUE to compute a MonteCarlo
# approximation of the predicted value distribution
# (quite long with ndraws=2000 by default)
\dontrun{
pred1MC <- predictY(m,newdata,draws=TRUE,var.time="Time")
}

## End(Not run)
```

---

print.lcmm — *Brief summary of a* `hlme`, `lcmm`, `Jointlcmm`,`multlcmm`, `epoce` *or* `Diffepoce` *objects*

Description

The function provides a brief summary of `hlme`, `lcmm`,`multlcmm` or `Jointlcmm` estimations, and epoce or `Diffepoce` computations.

Usage

```
## S3 method for class 'hlme'
print(x,...)
```

```
## S3 method for class 'Jointlcmm'
print(x,...)
## S3 method for class 'epoce'
print(x,...)
## S3 method for class 'lcmm'
print(x,...)
## S3 method for class 'Diffepoce'
print(x,...)
## S3 method for class 'multlcmm'
print(x,...)
```

Arguments

x	an object inheriting from classes hlme, lcmm for fitted latent class mixed-effects, or class Jointlcmmfor a Joint latent class mixed model or epoce for predictive accuracy computations.
...	further arguments to be passed to or from other methods. They are ignored in this function.

Author(s)

Cecile Proust-Lima, Viviane Philipps, Amadou Diakite and Benoit Liquet

See Also

hlme, lcmm, Jointlcmm, epoce, Diffepoce

---

summary.lcmm *Summary of a* hlme, lcmm, Jointlcmm, multlcmm, epoce *or* Diffepoce *objects*

Description

The function provides a summary of hlme, lcmm, multlcmm and Jointlcmm estimations, or epoce and Diffepoce computations.

Usage

```
## S3 method for class 'hlme'
summary(object,...)
## S3 method for class 'lcmm'
summary(object,...)
## S3 method for class 'Jointlcmm'
summary(object,...)
## S3 method for class 'epoce'
summary(object,...)
## S3 method for class 'Diffepoce'
```

```
summary(object,...)
## S3 method for class 'multlcmm'
summary(object,...)
```

Arguments

object	an object inheriting from classes hlme, lcmm, multlcmm for fitted latent class mixed-effects, or class Jointlcmm for a Joint latent class mixed model or epoce or Diffepoce for predictive accuracy computations.
...	further arguments to be passed to or from other methods. They are ignored in this function.

Value

For epoce or Diffepoce objects, returns NULL. For hlme, lcmm, Jointlcmm or multlcmm returns also a matrix containing the fixed effect estimates in the longitudinal model, their standard errors, Wald statistics and p-values

Author(s)

Cecile Proust-Lima, Viviane Philipps, Amadou Diakite and Benoit Liquet

See Also

hlme, lcmm, multlcmm, Jointlcmm, epoce, Diffepoce

---

summarytable *Summary of models*

Description

This function provides a table summarizing the results of different models fitted by hlme, lcmm, multlcmm or Jointlcmm.

Usage

```
summarytable(m1,...)
```

Arguments

m1	an object of class hlme, lcmm, multlcmm or Jointlcmm
...	further arguments, in particular other objects of class hlme, lcmm, multlcmm or Jointlcmm

Value

a matrix giving for each model: the number a latent classes, the log-likelihood, the number of parameters, the BIC and the posterior probability of the latent classes.

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

summary, hlme, lcmm, multlcmm, Jointlcmm

VarCov *Variance-covariance of the estimates*

Description

This function provides the variance-covariance matrix of the estimates. vcov is an alias for it.

Usage

```
## S3 method for class 'hlme'
VarCov(x)
## S3 method for class 'lcmm'
VarCov(x)
## S3 method for class 'Jointlcmm'
VarCov(x)
## S3 method for class 'multlcmm'
VarCov(x)
```

Arguments

x	an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`

Value

a matrix containing the variance-covariance of the estimates. For the parameters of the matrix of variance-covariance of the random effects, the Cholesky transformed parameters are considered so that VarCov provides the covariance matrix of function `estimates` with cholesky=TRUE.

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

estimates, hlme, lcmm, multlcmm, Jointlcmm

`VarCovRE` *Estimates, standard errors and Wald test for the parameters of the variance-covariance matrix of the random effects.*

Description

Fromm the Cholesky transformed parameters, this function provides estimates, standard errors and Wald test for the parameters of the variance-covariance matrix of the random effects.

Usage

```
## S3 method for class 'hlme'
VarCovRE(Mod)
## S3 method for class 'lcmm'
VarCovRE(Mod)
## S3 method for class 'Jointlcmm'
VarCovRE(Mod)
## S3 method for class 'multlcmm'
VarCovRE(Mod)
```

Arguments

`Mod`	an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`

Value

a matrix containing the estimates of the parameters of the variance-covariance matrix of the random effects, their standard errors, and, for the covariance parameters, the Wald statistic and the associated p-value.

Author(s)

Cecile Proust-Lima, Lionelle Nkam and Viviane Philipps

`VarExpl` *Percentage of variance explained by the (latent class) linear mixed model regression*

Description

The function provides the percentage of variance explained by the (latent class) linear mixed regression in a model estimated with `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`.

Usage

```
## S3 method for class 'hlme'
VarExpl(x,values)
## S3 method for class 'lcmm'
VarExpl(x,values)
## S3 method for class 'Jointlcmm'
VarExpl(x,values)
## S3 method for class 'multlcmm'
VarExpl(x,values)
```

Arguments

x	an object of class hlme, lcmm, multlcmm or Jointlcmm
values	a data frame with a unique row that contains the values of the variables in random and the time variable in the correlation process from which the percentage of variance should be calculated.

Value

For hlme, lcmm, and Jointlcmm objects, the function returns a matrix with 1 row and ng (ie the number of latent classes) columns containing (the class specific) percentages of variance explained by the linear mixed regression.

For multlcmm objects, the function returns a matrix containing (the class specific) percentages of variance explained by the linear mixed regression for each outcome. The resulting matrix is composed of as many rows as outcomes and as many columns as latent classes.

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

hlme, lcmm, multlcmm, Jointlcmm

Examples

```
## Not run:
m1 <- multlcmm(Ydep1+Ydep2~1+Time*X2+contrast(X2),random=~1+Time,
subject="ID",randomY=TRUE,link=c("4-manual-splines","3-manual-splines"),
intnodes=c(8,12,25),data=data_lcmm,
B=c(-1.071, -0.192,  0.106, -0.005, -0.193,  1.012,  0.870,  0.881,
  0.000,  0.000, -7.520,  1.401,  1.607 , 1.908,  1.431,  1.082,
 -7.528,  1.135 , 1.454 , 2.328, 1.052))

# variation percentages explained by linear mixed regression
VarExpl(m1,data.frame(Time=0))

## End(Not run)
```

WaldMult *Multivariate Wald Test*

Description

This function provides multivariate and univariate Wald tests for combinations of parameters from `hlme`, `lcmm`, `multlcmm` or `Jointlcmm` models.

Usage

```
WaldMult(Mod,pos=NULL,contrasts=NULL,name=NULL,value=NULL)
```

Arguments

`Mod`	an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`
`pos`	a vector containing the indices in Mod\$best of the parameters to test
`contrasts`	a numeric vector of same length as pos. If NULL (the default), a simultaneous test of the appropriate parameters is realised. If contrasts is specified, the quantity to test is the dot product of pos and contrasts.
`name`	a character containing the name the user wants to give to the test. By default, the name's test is the null hypothesis.
`value`	the value(s) to test against. By default, test against 0.

Value

If contrasts is NULL, the function returns a matrix with 1 row and 2 columns containing the value of the Wald test's statistic and the associated p-value.

If contrasts is not NULL, the function returns a matrix with 1 row and 4 columns containing the value of the coefficient (dot product of pos and contrasts), his standard deviation, the value of the Wald test's statistic and the associated p-value.

Author(s)

Cecile Proust-Lima, Lionelle Nkam and Viviane Philipps

Index