Package: lcmm (via r-universe)

October 28, 2026

Type Package

Title Extended Mixed Models Using Latent Classes and Latent Processes

Version 2.2.2

Date 2025-11-20

Description Estimation of various extensions of the mixed models including latent class mixed models, joint latent class mixed models, mixed models for curvilinear outcomes, mixed models for multivariate longitudinal outcomes using a maximum likelihood estimation method (Proust-Lima, Philipps, Liquet (2017) <doi:10.18637/jss.v078.i02>).

License GPL (>= 2.0)

Depends R (>= 3.5.0)

Imports nlme, survival (>= 2.37-2), parallel, mvtnorm, spacefillr, marqLevAlg (> 2.0), doParallel, numDeriv

Suggests knitr,rmarkdown,lattice,NormPsy

URL https://cecileproust-lima.github.io/lcmm/

BugReports https://github.com/CecileProust-Lima/lcmm/issues

LazyLoad yes

LazyData true

VignetteBuilder knitr

NeedsCompilation yes

RoxygenNote 7.3.3

Author Cecile Proust-Lima [aut, cre], Viviane Philipps [aut], Amadou Diakite [ctb], Benoit Liquet [ctb], Alan Genz [ctb] (Author of hrmsym subroutine), John Burkardt [ctb] (Author of alnorm subroutine)

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

Repository https://cran.r-universe.dev

Date/Publication 2025-11-20 18:46:35 UTC

RemoteUrl https://github.com/cran/lcmm

RemoteRef HEAD

RemoteSha f55258480e937c138a945e247d9f88f939def90a

Contents

lcmm-package . . . 3
createVarianceMatrix . . . 5
cuminc . . . 6
data_hlme . . . 7
data_lcmm . . . 8
Diffepoce . . . 9
dynpred . . . 11
epoce . . . 13
estimates . . . 15
externVar . . . 16
fitY . . . 23
ForInternalUse . . . 24
gridsearch . . . 25
hlme . . . 26
ItemInfo . . . 32
Jointlcmm . . . 34
lcmm . . . 44
loglik . . . 53
mpjlcmm . . . 60
multlcmm . . . 65
paquid . . . 74
permut . . . 75
plot . . . 76
plot.cuminc . . . 80
plot.Diffepoce . . . 81
plot.dynpred . . . 82
plot.ItemInfo . . . 84
plot.predict . . . 85
postprob . . . 87
predictClass . . . 88
predictCor . . . 89
predictL . . . 90
predictlink . . . 92
predictRE . . . 93
predictY . . . 94
predictYback . . . 98
predictYcond . . . 100
print.lcmm . . . 102
sampleParameters . . . 102
simdataHADS . . . 103
simulate.lcmm . . . 104

StandardMethods 106
summary.lcmm 107
summaryplot 108
summarytable 109
update.mpjlcmm 110
VarCov 111
VarCovRE 111
VarExpl 112
WaldMult 113
xclass 114

Index **115**

`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 for longitudinal and survival data (JLCM) and latent process mixed models (with or without latent classes of trajectory) for univariate and multivariate longitudinal outcomes of different types including curvilinear and ordinal outcomes. 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:	2.2.2
Date:	2025-11-20
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 (ordinal/binary) longitudinal outcomes using `lcmm` function,
- mixed models (with and without latent classes) for multivariate longitudinal outcomes of different nature using `multlcmm` function (this includes a longitudinal IRT model for homogeneous and heterogeneous data),
- 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,

- joint latent class mixed models for multivariate longitudinal outcomes and a right-censored (potentially left-truncated and of multiple causes) time-to-event using `mpjlcmm` function.

Please report any bug or comment regarding the package for future updates VIA GITHUB ONLY.

Author(s)

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

<cecile.proust-lima@u-bordeaux.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. doi:10.18637/jss.v078.i02

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

Proust-Lima, Philipps, Perrot, Blanchin, Sebille (2021). Modeling repeated self-reported outcome data: a continuous-time longitudinal Item Response Theory model. arXiv:210913064. http://arxiv.org/abs/2109.13064

Proust-Lima, Dartigues, Jacqmin-Gadda (2016). Joint modeling of repeated multivariate cognitive measures and competing risks of dementia and death: a latent process and latent class approach. Stat Med;35(3):382-98

Proust-Lima, Philipps, Dartigues, Bennett, Glymour, Jacqmin-Gadda, et al (2019). Are latent variable models preferable to composite score approaches when assessing risk factors of change? Evaluation of type-I error and statistical power in longitudinal cognitive studies. Stat Methods Med Res;28(7):1942-57

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

See Also

Useful links:

- https://cecileproust-lima.github.io/lcmm/
- Report bugs at https://github.com/CecileProust-Lima/lcmm/issues

---

`createVarianceMatrix` *Compute the variance matrix*

Description

The function computes the variance matrix of the random effects, the correlations, or the measurement error.

Usage

```
createVarianceMatrix(model, which, times1, times2, nmes)
```

Arguments

`model`	an object inheriting from class `hlme`, `lcmm`, `Jointlcmm` or `multlcmm` representing a general latent class mixed model.
`which`	either "random", "cor", "randomY", or "error".
`times1`	for `which = "cor"`, numeric vector containing the times at which the correlation should be computed
`times2`	for `which = "cor"`, numeric vector containing the times at which the correlation should be computed
`nmes`	for `which = "randomY"` and `which = "error"`, the number of measures (ie, the dimension of the returned variance matrix)

Value

a matrix

Author(s)

Viviane Philipps

Examples

```
## Not run:
 ## The model
 m <- hlme(fixed = Y ~ Time, mixture = ~1, random = ~1, subject = "ID",
  ng = 2, cor = BM(Time), data = data_hlme, B = c(0,20,30,-1,5,2,0.1))

## The random effects' variance matrix (the variance of the random intercept )
B <- createVarianceMatrix(m, which = "random")
```

```
## The variance of the Brownian motion at time c(1, 2, 3, 4)
W <- createVarianceMatrix(m, which = "cor", times1 = c(1, 2, 3, 4), times2 = c(1, 2, 3, 4))

## The variance of the measurement error at 4 visit times
S <- createVarianceMatrix(m, which = "error", nmes = 4)

## In model "m", the variance matrix of the outcome at times c(1, 2, 3, 4) is:
matrix(1, nrow = 4) %*% B %*% t(matrix(1, nrow = 4)) + W + S

## End(Not run)
```

`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, integrateOptions = NULL, ...)
```

Arguments

`x`	an object inheriting from class `Jointlcmm` or `mpjlcmm`
`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.
`integrateOptions`	optional list specifying the subdivisions, rel.tol and stop.on.error options (see ?integrate).
`...`	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

Examples

```
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))

par(mfrow=c(1,2))
plot(cuminc(m2,time=seq(0,20),X1=0,X2=0), ylim=c(0,1))
plot(cuminc(m2,time=seq(0,20),X1=0,X2=1), ylim=c(0,1))
```

---

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.

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

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.

Usage

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

Arguments

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

Details

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

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).

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

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

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, Jointlcmm, mpjlcmm, externSurv, or externX.

Usage

```
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

---

externVar — *Estimation of a secondary regression model after the estimation of a primary latent class model*

Description

This function fits regression models to relate a latent class structure (stemmed from a latent class model estimated within lcmm package) with either an external outcome or external class predictors. Two inference techniques are implemented. They both account for the classification error in the posterior class assignment:

- a 2-stage estimation using the joint likelihood of the primary latent class model and of the secondary/ external regression;

- a conditional regression of the external outcome given the underlying latent class structure, or of the underlying class structure given external covariates.

It returns an object of one of the lcmm package classes.

Usage

```
externVar(
  model,
  fixed,
  mixture,
  random,
  subject,
  classmb,
  survival,
  hazard = "Weibull",
  hazardtype = "Specific",
```

```
  hazardnodes = NULL,
  TimeDepVar = NULL,
  logscale = FALSE,
  idiag = FALSE,
  nwg = FALSE,
  randomY = NULL,
  link = NULL,
  intnodes = NULL,
  epsY = NULL,
  cor = NULL,
  nsim = NULL,
  range = NULL,
  data,
  longitudinal,
  method,
  varest,
  M = 200,
  B,
  convB = 1e-04,
  convL = 1e-04,
  convG = 1e-04,
  maxiter = 100,
  posfix,
  partialH = FALSE,
  verbose = FALSE,
  nproc = 1
)
```

Arguments

model	an object inheriting from class hlme, lcmm, Jointlcmm, multlcmm or mpjlcmm giving the primary latent class model.
fixed	optional, for secondary analyses on an external outcome variable: two-sided linear formula object for specifying the outcome and fixed-effect part in the secondary model. The response outcome is on the left of ~ and the covariates are separated by + on the right of the ~. The right side should be ~1 to model the outcome according to the latent classes only.
mixture	optional, for secondary analyses on an external outcome variable: one-sided formula object for the class-specific fixed effects in the model for the external outcome. 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, for secondary analyses on an external outcome variable: one-sided linear formula object for specifying the random effects in the secondary model, if appropriate. By default, no random effect is included.
subject	name of the covariate representing the grouping structure. Even in the absence of a hierarchical structure.

`classmb` optional, for secondary analyses on latent class membership according to external covariates: optional one-sided formula specifying the external predictors of latent class membership to be modeled in the secondary class-membership multinomial logistic model. Covariates are separated by + on the right of the ~.

`survival` optional, for secondary analyses on an external survival outcome: two-sided formula specifying the external survival part of the model. The right side should be ~1 to get the survival associated to each latent class without any other covariate.

`hazard` optional, for secondary analyses on an external survival outcome: family of hazard function assumed for the survival model (Weibull, piecewise or splines)

`hazardtype` optional, for secondary analyses on an external survival outcome: indicator for the type of baseline risk function (Specific, PH or Common)

`hazardnodes` optional, for secondary analyses on an external survival outcome: vector containing interior nodes if `splines` or `piecewise` is specified for the baseline hazard function in `hazard`

`TimeDepVar` optional, for secondary analyses on an external survival outcome: vector specifying the name of the time-dependent covariate in the survival model (only a irreversible event time in allowed)

`logscale` optional, for secondary analyses on an external survival outcome: 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

`idiag` optional, for secondary analyses on an external outcome: if appropriate, logical for the structure of the variance-covariance matrix of the random-effects in the secondary model. 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, for secondary analyses on an external outcome: if appropriate, logical indicating if the variance-covariance of the random-effects in the secondary model 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).

`randomY` optional, for secondary analyses on an external outcome: if appropriate, logical for including an outcome-specific random intercept. If FALSE no outcome-specific random intercept is added (default). If TRUE independent outcome-specific random intercept with parameterized variance are included

`link` optional, for secondary analyses on an external outcome: if appropriate, family of parameterized link functions for the external outcome if appropriate. Defaults to NULL, corresponding to continuous Gaussian distribution (hlme function).

`intnodes` optional, for secondary analyses on an external outcome: if appropriate, vector of interior nodes. This argument is only required for a I-splines link function with nodes entered manually.

`epsY` optional, for secondary analyses on an external outcome: if appropriate, 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, for secondary analyses on an external outcome: if appropriate, indicator for inclusion of an auto correlated 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.
nsim	optional, for secondary analyses on an external outcome: if appropriate, number of points to be used in the estimated link function. By default, nsom=100.
range	optional, for secondary analyses on an external outcome: if appropriate, 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.
data	Data frame containing the variables named in `fixed`, `mixture`, `random`, `classmb` and `subject`, for both the current function arguments and the primary model arguments Check `details` to get information on the data structure, especially with external outcomes.
longitudinal	only with `mpjlcmm` primary models and "twoStageJoint" method: mandatory list containing the longitudinal submodels used in the primary latent class model.
method	character indicating the inference technique to be used: `"twoStageJoint"` corresponds to 2-stage estimation using the joint log-likelihood. `"conditional"` corresponds to the conditional regression using the underlying true latent class membership.
varest	optional character indicating the method to be used to compute the variance of the regression estimates in the secondary regression. `"none"` does not account for the uncertainty in the primary latent class model, `"paramBoot"` computes the total variance using a parametric bootstrap technique, `"Hessian"` computes the total Hessian of the joint likelihood (implemented for `"twoStageJoint"` method only). Default to `"Hessian"` for `"twoStageJoint"` method and `"paramBoot"` for `"conditional"` method.
M	option integer indicating the number of draws for the parametric boostrap when `varest="paramBoot"`. Default to 200.
B	optional vector of initial parameter values for the secondary model. With an external outcome, the vector has the same structure as a latent class model estimated in the other functions of `lcmm` package for the same type of outcome except that no parameters should be included for the latent class membership. With external class predictors (of size p), the vector is of length (ng-1)*(1+p). If `B=NULL` (by default), internal initial values are considered
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 secondary model estimation using Marquardt iterative algorithm. Defaults to 100
posfix	optional vector specifying indices in parameter vector B the secondary model that should not be estimated. Default to NULL, all the parameters of the secondary regression are estimated.
partialH	optional logical for Piecewise and Splines baseline risk functions and Splines link functions only. Indicates whether the parameters of the baseline risk or link functions can be dropped from the Hessian matrix to define convergence criteria (can solve non convergence due to estimates at the boundary of the parameter space - usually 0).
verbose	logical indicating whether information about computation should be reported. Default to FALSE.
nproc	the number cores for parallel computation. Default to 1 (sequential mode).

Details

A. DATA STRUCTURE

The data argument must follow specific structure. It must include all the data necessary to compute the posterior classification probabilities (so a longitudinal format usually) as well as the information for the secondary analysis. For time-invariant variables in the secondary analyses: - if used as an external outcome: the information should not be duplicated at each row of the subject. It should appear once for each individual. - if used as an external covariate: the information can be duplicated at each row of the subject (as usual)

B. VARIANCE ESTIMATION

The two techniques rely on a sequential analysis (two-stage analysis) so the variance calculation should account for both the uncertainty in the first and the second stage. Not taking into account the first-stage uncertainty by specifying varest="none" may lead to the underestimation of the final variance. When possible, Method varest="Hessian" which relies on the combination of Hessians from the primary and secondary models is recommended. However, it may become numerically intensive when the primary latent class model includes a high number of parameters. As an alternative, especially when the primary model is complex and the second model includes a limited number of parameters, the parametric Bootstrap method varest="paramBoot" can be favored.

Value

an object of class externVar and externSurv for external survival outcomes, externX for external class predictors, and hlme, lcmm, or multlcmm for external longitudinal or cross-sectional outcomes.

Author(s)

Maris Dussartre, Cecile Proust-Lima and Viviane Philipps

Examples

```
## Not run:


###### Estimation of the primary latent class model                  ######
```

```
# this is a linear latent class mixed model for Ydep1
# with 2 classes and a linear trajectory

set.seed(1234)
PrimMod <- hlme(Ydep1~Time,random=~Time,subject='ID',ng=1,data=data_lcmm)
PrimMod2 <- hlme(Ydep1~Time,mixture=~Time,random=~Time,subject='ID',
                 ng=2,data=data_lcmm,B=random(PrimMod))

###### Example 1: Relationship between the latent class structure and       #
#                    external class predictors                            ######

# We consider here 4 external predictors X1-X4.

# estimation of the secondary multinomial logistic model with total variance
# computed with the Hessian

XextHess <- externVar(PrimMod2,
                      classmb = ~X1 + X2 + X3 + X4,
                      subject = "ID",
                      data = data_lcmm,
                      method = "twoStageJoint")
summary(XextHess)

# estimation of a secondary multinomial logistic model with total variance
# computed with parametric Bootstrap (much longer). When planning to use
# the bootstrap estimator, we recommend running first the analysis
# with option varest = "none" which is faster but which underestimates
# the variance. And then use these values as plausible initial values when
# running the estimation with varest = "paramBoot" to obtain  a valid
# variance of the parameters.

XextNone <- externVar(PrimMod2,
                      classmb = ~X1 + X2 + X3 + X4,
                      subject = "ID",
                      data = data_lcmm,
                      varest = "none",
                      method = "twoStageJoint")

XextBoot <- externVar(PrimMod2,
                      classmb = ~X1 + X2 + X3 + X4,
                      subject = "ID",
                      data = data_lcmm,
                      varest = "paramBoot",
                      method = "twoStageJoint",
                      B = XextNone$best)
summary(XextBoot)


###### Example 2: Relationship between a latent class structure and         #
#                external outcome (repeatedly measured over time)     ######


# We want to estimate a linear mixed model for Ydep2 with a linear trajectory
```

```
# adjusted on X1.

# estimation of the secondary linear mixed model with total variance
# computed with the Hessian

YextHess = externVar(PrimMod2,   #primary model
                     fixed = Ydep2 ~ Time*X1,  #secondary model
                     random = ~Time, #secondary model
                     mixture = ~Time,  #secondary model
                     subject="ID",
                     data=data_lcmm,
                     method = "twoStageJoint")


# estimation of a secondary linear mixed model with total variance
# computed with parametric Bootstrap (much longer). When planning to use
# the bootstrap estimator, we recommend running first the analysis
# with option varest = "none" which is faster but which underestimates
# the variance. And then use these values as plausible initial values when
# running the estimation with varest = "paramBoot" to obtain  a valid
# variance of the parameters.

YextNone = externVar(PrimMod2,   #primary model
                     fixed = Ydep2 ~ Time*X1,  #secondary model
                     random = ~Time, #secondary model
                     mixture = ~Time,  #secondary model
                     subject="ID",
                     data=data_lcmm,
                     varest = "none",
                     method = "twoStageJoint")

YextBoot = externVar(PrimMod2,   #primary model
                     fixed = Ydep2 ~ Time*X1,  #secondary model
                     random = ~Time, #secondary model
                     mixture = ~Time,  #secondary model
                     subject="ID",
                     data=data_lcmm,
                     method = "twoStageJoint",
                     B = YextNone$best,
                     varest= "paramBoot")

summary(YextBoot)


###### Example 3: Relationship between a latent class structure and         #
#                      external outcome (survival)                     ######

# We want to estimate a proportional hazard model (with proportional hazard
# across classes) for time to event Tevent (indicator Event) and assuming
# a splines baseline risk with 3 knots.

# estimation of the secondary survival model with total variance
# computed with the Hessian
```

```
YextHess = externVar(PrimMod2,   #primary model
                     survival = Surv(Tevent,Event)~ X1+mixture(X2), #secondary model
                     hazard="3-quant-splines", #secondary model
                     hazardtype="PH", #secondary model
                     subject="ID",
                     data=data_lcmm,
                     method = "twoStageJoint")
summary(YextHess)


# estimation of a secondary survival model with total variance
# computed with parametric Bootstrap (much longer). When planning to use
# the bootstrap estimator, we recommend running first the analysis
# with option varest = "none" which is faster but which underestimates
# the variance. And then use these values as plausible initial values when
# running the estimation with varest = "paramBoot" to obtain  a valid
# variance of the parameters.

YextNone = externVar(PrimMod2,   #primary model
                     survival = Surv(Tevent,Event)~ X1+mixture(X2), #secondary model
                     hazard="3-quant-splines", #secondary model
                     hazardtype="PH", #secondary model
                     subject="ID",
                     data=data_lcmm,
                     varest = "none",
                     method = "twoStageJoint")

YextBoot = externVar(PrimMod2,   #primary model
                     survival = Surv(Tevent,Event)~ X1+mixture(X2), #secondary model
                     hazard="3-quant-splines", #secondary model
                     hazardtype="PH", #secondary model
                     subject="ID",
                     data=data_lcmm,
                     method = "twoStageJoint",
                     B = YextNone$best,
                     varest= "paramBoot")

summary(YextBoot)


## End(Not run)
```

---

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

```
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, cl = NULL)
```

Arguments

m	a call of hlme, lcmm, multlcmm, Jointlcmm or mpjlcmm 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, Jointlcmm or mpjlcmm 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
cl	a cluster created by makeCluster from package parallel or an integer specifying the number of cores to use for parallel computation

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, Jointlcmm or mpjlcmm corresponding to the call specified in m.

Author(s)

Cecile Proust-Lima and Viviane Philipps

References

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 = 1e-04,
  convL = 1e-04,
  convG = 1e-04,
  prior,
  pprior = NULL,
  maxiter = 500,
  subset = NULL,
  na.action = 1,
  posfix = NULL,
  verbose = FALSE,
  returndata = FALSE,
  var.time = NULL,
  partialH = FALSE,
  nproc = 1,
  clustertype = NULL
)
```

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 +. By default, classmb=~1 if ng>1.
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.

`pprior` optional vector specifying the names of the covariates containing the prior probabilities to belong to each latent class. These probabilities should be between 0 and 1 and should sum up to 1 for each subject.

`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.

`returndata` logical indicating if data used for computation should be returned. Default to FALSE, data are not returned.

var.time	optional character indicating the name of the time variable.
partialH	optional logical indicating if parameters can be dropped from the Hessian matrix to define convergence criteria.
nproc	the number cores for parallel computation. Default to 1 (sequential mode).
clustertype	optional character indicating the type of cluster for parallel computation.

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`	if the model converged (conv=1 or 3), 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`. If conv=2, `V` contains the second derivatives of the log-likelihood.
`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, =3 if the convergence criteria were satisfied with a partial Hessian matrix, =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; 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,...). If `var.time` is specified, the corresponding measurement time is also included.
`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
`data`	the original data set (if returndata is TRUE)

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. doi:10.18637/jss.v078.i02

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")
```

`ItemInfo` *Conditional probabilities and item information given specified latent process values for* `lcmm` *or* `multlcmm` *object with ordinal outcomes.*

Description

The function computes the conditional probability and information function of each level of each ordinal outcome and the information function at the item level. Confidence bands (and median) can be computed by a Monte Carlo approximation.

Usage

```
ItemInfo(
```

```
  x,
  lprocess,
  condRE_Y = FALSE,
  nsim = 200,
  draws = FALSE,
  ndraws = 2000,
  ...
)
```

Arguments

x	an object inheriting from class lcmm or multlcmm, representing a general (latent class) mixed model.
lprocess	numeric vector containing the latent process values at which the predictions should be computed.
condRE_Y	for multlcmm objects only, logical indicating if the predictions are conditional to the outcome-specific random-effects or not. Default to FALSE= the predictions are marginal to these random effects.
nsim	number of points used in the numerical integration (Monte-Carlo) with splines or Beta link functions. nsim should be relatively important (nsim=200 by default).
draws	optional boolean specifying whether median and confidence bands of the predicted values should be computed (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.
...	further arguments to be passed to or from other methods. They are ignored in this function.

Value

An object of class ItemInfo with values :

- ItemInfo: If draws=FALSE, returns a matrix with 3 columns: the first column indicates the name of the outcome, the second indicates the latent process value and the last is the computed Fisher information. If draws=TRUE, returns a matrix with 5 columns: the name of the outcome, the latent process value and the 50%, 2.5% and 97.5% percentiles of the approximated posterior distribution of information.

- LevelInfo: If draws=FALSE, returns a matrix with 5 columns: the first column indicates the name of the outcome, the second indicates the outcome's level, the third indicates the latent process value and the two last contain the probability and Fisher information. If draws=TRUE, returns a matrix with 5 columns: the name of the outcome, the outcome's level, the latent process value and the 50%, 2.5% and 97.5% percentiles of the approximated posterior distribution of the probability and information.

- object: the model from which the computations are done.

- IC: indicator specifying if confidence intervals are computed.

Author(s)

Cecile Proust-Lima, Viviane Philipps

Examples

```
## Not run:
## This is a toy example to illustrate the information functions.
## The binary outcomes are arbitrarily created, please do not
## consider them as relevent indicators.
data_lcmm$Yord1 <- as.numeric(data_lcmm$Ydep1>10)
data_lcmm$Yord2 <- as.numeric(data_lcmm$Ydep2>25)
m <- multlcmm(Yord1+Yord2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="thresholds")
info <- ItemInfo(m,lprocess=seq(-4,4,length.out=100),draws=TRUE)
plot(info)
par(mfrow=c(1,2))
plot(info, which="LevelInfo", outcome="Yord1")
plot(info, which="LevelInfo", outcome="Yord2")
plot(info, which="LevelProb", outcome="Yord1")
plot(info, which="LevelProb", outcome="Yord2")

## End(Not run)
```

`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,
  hazardrange = NULL,
  TimeDepVar = NULL,
  link = NULL,
  intnodes = NULL,
  epsY = 0.5,
  range = NULL,
  cor = NULL,
  data,
  B,
  convB = 1e-04,
  convL = 1e-04,
  convG = 1e-04,
  maxiter = 100,
  nsim = 100,
  prior = NULL,
  pprior = NULL,
  logscale = FALSE,
  subset = NULL,
  na.action = 1,
  posfix = NULL,
  partialH = FALSE,
  verbose = FALSE,
  returndata = FALSE,
  var.time = NULL,
  nproc = 1,
  clustertype = NULL
)

jlcmm(
  fixed,
  mixture,
  random,
  subject,
  classmb,
  ng = 1,
  idiag = FALSE,
  nwg = FALSE,
  survival,
  hazard = "Weibull",
  hazardtype = "Specific",
  hazardnodes = NULL,
  hazardrange = NULL,
  TimeDepVar = NULL,
  link = NULL,
  intnodes = NULL,
  epsY = 0.5,
```

```
  range = NULL,
  cor = NULL,
  data,
  B,
  convB = 1e-04,
  convL = 1e-04,
  convG = 1e-04,
  maxiter = 100,
  nsim = 100,
  prior = NULL,
  pprior = NULL,
  logscale = FALSE,
  subset = NULL,
  na.action = 1,
  posfix = NULL,
  partialH = FALSE,
  verbose = FALSE,
  returndata = FALSE,
  var.time = NULL,
  nproc = 1,
  clustertype = NULL
)
```

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.

hazardrange — optional vector indicating the range of the survival times (that is the minimum and maximum). By default, the range is defined according to the minimum and maximum observed values of the survival times. The option should be used only for piecewise constant and Splines hazard functions.

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 Jointlcmm 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.
pprior	optional vector specifying the names of the covariates containing the prior probabilities to belong to each latent class. These probabilities should be between 0 and 1 and should sum up to 1 for each subject.
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 and Splines link functions only. Indicates whether the parameters of the baseline risk or 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.
returndata	logical indicating if data used for computation should be returned. Default to FALSE, data are not returned.
var.time	optional character indicating the name of the time variable.
nproc	the number cores for parallel computation. Default to 1 (sequential mode).
clustertype	optional character indicating the type of cluster for parallel computation.

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`	if the model converged (conv=1 or 3), 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`. If conv=2, `V` contains the second derivatives of the log-likelihood.
`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,...). If `var.time` is specified, the corresponding measurement time is also included.
`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
data	the original data set (if returndata is TRUE)

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. doi:10.18637/jss.v078.i02

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

```
## Not run:
#### 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.


#### 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,
  pprior = NULL,
  range = NULL,
  subset = NULL,
  na.action = 1,
  posfix = NULL,
  partialH = FALSE,
  verbose = FALSE,
  returndata = FALSE,
  var.time = NULL,
  nproc = 1,
  clustertype = NULL,
  computeDiscrete = NULL
)
```

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, no random effect is 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).
pprior	optional vector specifying the names of the covariates containing the prior probabilities to belong to each latent class. These probabilities should be between 0 and 1 and should sum up to 1 for each subject.
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 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.
returndata	logical indicating if data used for computation should be returned. Default to FALSE, data are not returned.
var.time	optional character indicating the name of the time variable.
nproc	the number cores for parallel computation. Default to 1 (sequential mode).

`clustertype` optional character indicating the type of cluster for parallel computation.

`computeDiscrete` optional logical indicating if a dscrete likelihood and UACV should be computed. By default, if the outcome only consists of integers computeDiscrete=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`	if the model converged (conv=1 or 3), 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`. If conv=2, `V` contains the second derivatives of the log-likelihood.
`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,...). If `var.time` is specified, the corresponding measurement time is also included. 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
data	the original data set (if returndata is TRUE)

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. doi:10.18637/jss.v078.i02

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)
```

`loglik` *Wrapper to the Fortran subroutines computing the log-likelihood*

Description

Log-likelihood of hlme, lcmm, multlcmm, Jointlcmm and mpjlcmm models. The argument's specification is not straightforward, so these functions are usually not directly used.

Usage

```
loglikhlme(
```

```
  b,
  Y0,
  X0,
  prior0,
  pprior0,
  idprob0,
  idea0,
  idg0,
  idcor0,
  ns0,
  ng0,
  nv0,
  nobs0,
  nea0,
  nmes0,
  idiag0,
  nwg0,
  ncor0,
  npm0,
  fix0,
  nfix0,
  bfix0
)

logliklcmm(
  b,
  Y0,
  X0,
  prior0,
  pprior0,
  idprob0,
  idea0,
  idg0,
  idcor0,
  ns0,
  ng0,
  nv0,
  nobs0,
  nea0,
  nmes0,
  idiag0,
  nwg0,
  ncor0,
  npm0,
  epsY0,
  idlink0,
  nbzitr0,
  zitr0,
```

```
  minY0,
  maxY0,
  ide0,
  fix0,
  nfix0,
  bfix0
)

loglikmultlcmm(
  b,
  Y0,
  X0,
  prior0,
  pprior0,
  idprob0,
  idea0,
  idg0,
  idcor0,
  idcontr0,
  ny0,
  ns0,
  ng0,
  nv0,
  nobs0,
  nea0,
  nmes0,
  idiag0,
  nwg0,
  ncor0,
  nalea0,
  npm0,
  epsY0,
  idlink0,
  nbzitr0,
  zitr0,
  uniqueY0,
  indiceY0,
  nvalSPLORD0,
  fix0,
  nfix0,
  bfix0,
  methInteg0,
  nMC0,
  dimMC0,
  seqMC0,
  chol0
)
```

```
loglikJointlcmm(
  b,
  Y0,
  X0,
  prior0,
  pprior0,
  tentr0,
  tevt0,
  devt0,
  ind_survint0,
  idprob0,
  idea0,
  idg0,
  idcor0,
  idcom0,
  idspecif0,
  idtdv0,
  idlink0,
  epsY0,
  nbzitr0,
  zitr0,
  uniqueY0,
  nvalSPL0,
  indiceY0,
  typrisq0,
  risqcom0,
  nz0,
  zi0,
  ns0,
  ng0,
  nv0,
  nobs0,
  nmes0,
  nbevt0,
  nea0,
  nwg0,
  ncor0,
  idiag0,
  idtrunc0,
  logspecif0,
  npm0,
  fix0,
  nfix0,
  bfix0
)

loglikmpjlcmm(
  b,
```

```
    K0,
    ny0,
    nbevt0,
    ng0,
    ns0,
    Y0,
    nobs0,
    X0,
    nv0,
    Xns0,
    nv20,
    prior0,
    Tentr0,
    Tevt0,
    Devt0,
    ind_survint0,
    idnv0,
    idnv20,
    idspecif0,
    idlink0,
    epsY0,
    nbzitr0,
    zitr0,
    uniqueY0,
    nvalSPL0,
    indiceY0,
    typrisq0,
    risqcom0,
    nz0,
    zi0,
    nmes0,
    nea0,
    nw0,
    ncor0,
    nalea0,
    idiag0,
    idtrunc0,
    logspecif0,
    npm0,
    fix0,
    contrainte0,
    nfix0,
    bfix0
)
```

Arguments

b	the vector of estimated parameters (length npm0)

Y0	the observed values of the outcome(s) (length nobs0)
X0	the observed values of all covariates included in the model (dim nob0 * nv0)
prior0	the prior latent class (length ns0)
pprior0	the prior probabilty of each latent class (dim ns0 * ng0)
idprob0	indicator of presence in the class membership submodel (length nv0)
idea0	indicator of presence in the random part of the longitudinal submodel (length nv0)
idg0	indicator of presence in the fixed part of the longitudinal submodel (length nv0)
idcor0	indicator of presence in the correlation part of the longitudinal submodel (length nv0)
ns0	number of subjects
ng0	number of latent classes
nv0	number of covariates
nobs0	number of observations
nea0	number of random effects
nmes0	number of mesures for each subject (length ns0 or dom ns0*ny0)
idiag0	indicator of diagonal variance matrix of the random effects
nwg0	number of parameters for proportional random effects over latent classes
ncor0	number of parameters for the correlation
npm0	total number of parameters
fix0	indicator of non estimated parameter (length npm0+nfix0)
nfix0	number of non estimated parameters
bfix0	vector of non estimated parameters
epsY0	epsY values for Beta transformations
idlink0	type of transformation
nbzitr0	number of nodes for the transformations
zitr0	nodes for the transformations
minY0	minimum value for the longitudinal outcome
maxY0	maximum value for the longitudinal outcome
ide0	indicator of observed values for ordinal outcomes
idcontr0	indicator of presence as contrast in the fixed part of the longitudinal submodel (length nv0)
ny0	number of longitudinal outcomes
nalea0	number of parameters f the outcome specific random effect
uniqueY0	unique values of the longitudinal outcomes
indiceY0	correspondance between Y0 and uniqueY0
nvalSPLORD0	number of unique values for outcomes modeled with splines transformations or as ordinal outcome

methInteg0	type of integration
nMC0	number of nodes for Monte Carlo integration
dimMC0	dimension of the integration
seqMC0	sequence of integration nodes
chol0	indicator of Cholesky parameterization
tentr0	entry time for the survival submodel
tevt0	event time for the survival submodel
devt0	indicator of event for the survival submodel
ind_survint0	indicator of risk change
idcom0	indicator of presence in the survival submodel with common effect
idspecif0	indicator of presence in the survival submodel with cause-specific or class specific effect
idtdv0	indicator of 'TimeDepVar' covariate
nvalSPL0	number of unique values for outcomes modeled with splines transformations
typrisq0	type of baseline risk
risqcom0	specification of baseline risk across latent classes
nz0	number of nodes for the baseline
zi0	nodes for the baseline
nbevt0	number of events
idtrunc0	indicator of left truncation
logspecif0	indicator of logarithm parameterization
K0	number of latent processes
Xns0	the observed values of the covariates included in the survival submodel (dim ns0*nv20)
nv20	number of covariates in Xns0
Tentr0	entry time for the survival submodel (length ns0)
Tevt0	event time for the survival submodel (length ns0)
Devt0	indicator of event for the survival submodel (length ns0)
idnv0	indicator of presence in each subpart of the longitudinal models (length 4*sum(nv0))
idnv20	indicator of presence in each subpart of the survival models (length 3*nv20)
nw0	number of parameters for proportional random effects over latent classes
contrainte0	type of identifiability constraints

Value

the log-likelihood

Author(s)

Cecile Proust-Lima, Viviane Philipps

mpjlcmm *Estimation of multivariate joint latent class mixed models*

Description

This function fits joint latent class models for multivariate longitudinal markers and competing causes of event. It is a multivariate extension of the Jointlcmm function. It defines each longitudinal dimension as a latent process (mp in mpjlcmm is for multivariate processes), possibly measured by sereval continuous markers (Gaussian or curvilinear). For the moment, theses processes are assumed independent given the latent classes. The (optional) survival part handles competing risks, right censoring and left truncation. The specification of the function is similar to other estimating functions of the package.

Usage

```
mpjlcmm(
  longitudinal,
  subject,
  classmb,
  ng,
  survival,
  hazard = "Weibull",
  hazardtype = "Specific",
  hazardnodes = NULL,
  TimeDepVar = NULL,
  data,
  B,
  convB = 1e-04,
  convL = 1e-04,
  convG = 1e-04,
  maxiter = 100,
  nsim = 100,
  prior,
  logscale = FALSE,
  subset = NULL,
  na.action = 1,
  posfix = NULL,
  partialH = FALSE,
  verbose = FALSE,
  nproc = 1,
  clustertype = NULL
)
```

Arguments

longitudinal	list of longitudinal models of type hlme, lcmm or multlcmm. Each model defines the structure of one latent process.

subject	name of the covariate representing the grouping structure (called subject identifier)
classmb	optional one-sided formula describing the covariates in the class-membership multinomial logistic model
ng	number of latent classes considered
survival	two-sided formula object specifying the survival part of the model
hazard	optional family of hazard function assumed for the survival model (Weibull, piecewise or splines)
hazardtype	optional indicator for the type of baseline risk function (Specific, PH or Common)
hazardnodes	optional vector containing interior nodes if splines or piecewise is specified for the baseline hazard function in hazard
TimeDepVar	optional vector specifying the name of the time-dependent covariate in the survival model
data	data frame containing all the variables used in the model
B	optional specification for the initial values of 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. Initial values are extracted from the models specified in longitudinal, and default initial values are chosen for the survival part (3) when ng>1, a mpjlcmm 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. Note that due to possible local maxima, the B vector should be specified and several different starting points should be tried. This can be done automatically using gridsearch function.
convB	optional threshold for the convergence criterion based on the parameter stability
convL	optional threshold for the convergence criterion based on the log-likelihood stability
convG	optional threshold for the convergence criterion based on the derivatives
maxiter	optional maximum number of iterations for the Marquardt iterative algorithm
nsim	optional number of points for the predicted survival curves and predicted baseline risk curves
prior	optional name of a covariate containing a prior information about the latent class membership
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
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 and Splines link functions only. Indicates whether the parameters of the baseline risk or link functions can be dropped from the Hessian matrix to define convergence criteria (can solve non convergence due to estimates at the boundary of the parameter space - usually 0).
verbose	logical indicating whether information about computation should be reported. Default to FALSE.
nproc	the number cores for parallel computation. Default to 1 (sequential mode).
clustertype	optional character indicating the type of cluster for parallel computation.

Author(s)

Cecile Proust Lima and Viviane Philipps

Examples

```
## Not run:
paquid$age65 <- (paquid$age-65)/10

#############################################################################
###                          EXAMPLE 1 :                                  ###
###two outcomes measuring the same latent process along with dementia onset###
#############################################################################

## multlcmm model for MMSE and BVRT for 1 class
mult1 <- multlcmm(MMSE+BVRT~age65+I(age65^2)+CEP+male,random=~age65+I(age65^2),
subject="ID",link=c("5-quant-splines","4-quant-splines"),data=paquid)
summary(mult1)

## joint model for 1 class
m1S <- mpjlcmm(longitudinal=list(mult1),subject="ID",ng=1,data=paquid,
survival=Surv(age_init,agedem,dem)~1)
summary(m1S)


##### joint model for 2 classes #####

## specify longitudinal model for 2 classes, without estimation
mult2 <- multlcmm(MMSE+BVRT~age65+I(age65^2)+CEP+male,random=~age65+I(age65^2),
subject="ID",link=c("5-quant-splines","4-quant-splines"),ng=2,
mixture=~age65+I(age65^2),data=paquid,B=random(mult1),maxiter=0)

## estimation of the associated joint model
m2S <- mpjlcmm(longitudinal=list(mult2),subject="ID",ng=2,data=paquid,
survival=Surv(age_init,agedem,dem)~1)

## estimation by a grid search with 50 replicates, initial values
## randomly generated from m1S
```

```
m2S_b <- gridsearch(mpjlcmm(longitudinal=list(mult2),subject="ID",ng=2,
data=paquid,survival=Surv(age_init,agedem,dem)~1), minit=m1S, rep=50, maxiter=30)


##### joint model for 3 classes #####
mult3 <- multlcmm(MMSE+BVRT~age65+I(age65^2)+CEP+male,random=~age65+I(age65^2),
subject="ID",link=c("5-quant-splines","4-quant-splines"),ng=3,
mixture=~age65+I(age65^2),data=paquid,B=random(mult1),maxiter=0)

m3S <- mpjlcmm(longitudinal=list(mult3),subject="ID",ng=3,data=paquid,
survival=Surv(age_init,agedem,dem)~1)

m3S_b <- gridsearch(mpjlcmm(longitudinal=list(mult3),subject="ID",ng=3,
data=paquid,survival=Surv(age_init,agedem,dem)~1), minit=m1S, rep=50, maxiter=30)


##### summary of the models #####

summarytable(m1S,m2S,m2S_b,m3S,m3S_b)



##### post-fit #####

## update longitudinal models :
mod2 <- update(m2S)

mult2_post <- mod2[[1]]
## -> use the available functions for multlcmm on the mult2_post object

## fit of the longitudinal trajectories
par(mfrow=c(2,2))
plot(mult2_post,"fit","age65",marg=TRUE,shades=TRUE,outcome=1)
plot(mult2_post,"fit","age65",marg=TRUE,shades=TRUE,outcome=2)

plot(mult2_post,"fit","age65",marg=FALSE,shades=TRUE,outcome=1)
plot(mult2_post,"fit","age65",marg=FALSE,shades=TRUE,outcome=2)


## predicted trajectories
dpred <- data.frame(age65=seq(0,3,0.1),male=0,CEP=0)

predL <- predictL(mult2_post,newdata=dpred,var.time="age65",confint=TRUE)
plot(predL,shades=TRUE) # in the latent process scale


predY <- predictY(mult2_post,newdata=dpred,var.time="age65",draws=TRUE)

plot(predY,shades=TRUE,ylim=c(0,30),main="MMSE") #in the 0-30 scale for MMSE
plot(predY,shades=TRUE,ylim=c(0,15),outcome=2,main="BVRT") #in 0-15 for BVRT

## baseline hazard and survival curves :
plot(m2S,"hazard")
```

```
plot(m2S,"survival")

## posteriori probabilities and classification :
postprob(m2S)



#############################################################################################
###                                EXAMPLE 2 :                                            ###
### two latent processes measured each by one outcome along with dementia onset  ###
#############################################################################################

## define the two longitudinal models

mMMSE1 <- lcmm(MMSE~age65+I(age65^2)+CEP,random=~age65+I(age65^2),subject="ID",
link="5-quant-splines",data=paquid)

mCESD1 <- lcmm(CESD~age65+I(age65^2)+male,random=~age65+I(age65^2),subject="ID",
link="5-quant-splines",data=paquid)


## joint estimation

mm1S <- mpjlcmm(longitudinal=list(mMMSE1,mCESD1),subject="ID",ng=1,data=paquid,
survival=Surv(age_init,agedem,dem)~CEP+male)


## with 2 latent classes

mMMSE2 <- lcmm(MMSE~age65+I(age65^2)+CEP,random=~age65+I(age65^2),subject="ID",
link="5-quant-splines",data=paquid,ng=2,mixture=~age65+I(age65^2),
B=random(mMMSE1),maxiter=0)

mCESD2 <- lcmm(CESD~age65+I(age65^2)+male,random=~age65+I(age65^2),subject="ID",
link="5-quant-splines",data=paquid,ng=2,mixture=~age65+I(age65^2),
B=random(mCESD1),maxiter=0)

mm2S <- mpjlcmm(longitudinal=list(mMMSE2,mCESD2),subject="ID",ng=2,data=paquid,
survival=Surv(age_init,agedem,dem)~CEP+mixture(male),classmb=~CEP+male)

mm2S_b <- gridsearch(mpjlcmm(longitudinal=list(mMMSE2,mCESD2),subject="ID",ng=2,
data=paquid,survival=Surv(age_init,agedem,dem)~CEP+mixture(male),
classmb=~CEP+male),minit=mm1S,rep=50,maxiter=50)

summarytable(mm1S,mm2S,mm2S_b)


mod1_biv <- update(mm1S)
mod2_biv <- update(mm2S)

## -> use post-fit functions as in exemple 1

## End(Not run)
```

`multlcmm` *Estimation of multivariate mixed-effect models and multivariate latent class mixed-effect models for multivariate longitudinal outcomes of possibly multiple types (continuous Gaussian, continuous non-Gaussian/curvilinear, ordinal) 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 ordinal longitudinal outcomes (with cumulative probit measurement model). 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 functions `hlme` and `lcmm`). 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,
  pprior = NULL,
```

```
  range = NULL,
  subset = NULL,
  na.action = 1,
  posfix = NULL,
  partialH = FALSE,
  verbose = FALSE,
  returndata = FALSE,
  methInteg = "QMC",
  nMC = NULL,
  var.time = NULL,
  nproc = 1,
  clustertype = NULL
)

mlcmm(
  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,
  pprior = NULL,
  range = NULL,
  subset = NULL,
  na.action = 1,
  posfix = NULL,
  partialH = FALSE,
  verbose = FALSE,
  returndata = FALSE,
  methInteg = "QMC",
  nMC = NULL,
  var.time = NULL,
  nproc = 1,
```

```
  clustertype = NULL
)
```

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, "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 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).

`pprior` optional vector specifying the names of the covariates containing the prior probabilities to belong to each latent class. These probabilities should be between 0 and 1 and should sum up to 1 for each subject.

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 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.
returndata	logical indicating if data used for computation should be returned. Default to FALSE, data are not returned.
methInteg	character indicating the type of integration if ordinal outcomes are considered. 'MCO' for ordinary Monte Carlo, 'MCA' for antithetic Monte Carlo, 'QMC' for quasi Monte Carlo. Default to "QMC".
nMC	integer, number of Monte Carlo simulations. By default, 1000 points are used if at least one threshold link is specified.
var.time	optional character indicating the name of the time variable.
nproc	the number cores for parallel computation. Default to 1 (sequential mode).
clustertype	optional character indicating the type of cluster for parallel computation.

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.

4. With the "thresholds" link function for an ordinal outcome with levels 0,...,C, C-1 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. To constraint the parameters to be increasing, except for the first parameter b_1, the program estimates b_k^* (for k=2,...C-1) so that b_k=b_(k-1)+(b_k^*)^2.

Details of these parameterized link functions can be found in the papers: Proust-Lima et al. (Biometrics 2006), Proust-Lima et al. (BJMSP 2013), Proust-Lima et al. (arxiv 2021 - https://arxiv.org/abs/2109.13064)

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 intercept parameter should be specified in `B`. When ng>1, the first intercept is not estimated and only ng-1 intercept 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. (6) only if `nwg=TRUE` and ng>1, ng-1 parameters for class-specific proportional coefficients for the variance covariance matrix of the random-effects; (7) if `cor` is specified, the standard error of the Brownian motion or the standard error and the correlation parameter of the autoregressive process; (8) the standard error of the outcome-specific Gaussian errors (one per outcome); (9) if `randomY=TRUE`, the standard error of the outcome-specific random intercept (one per outcome); (10) 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 or 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

When dealing only with continuous outcomes, the computation of the likelihood does not require any numerical integration over the random-effects, so that the estimation procedure is relatively fast. When at least one ordinal outcome is modeled, a numerical integration over the random-effects is required in each computation of the individual contribution to the likelihood. This achieved using a Monte-Carlo procedure. We allow three options: the standard Monte-Carlo simulations, as well as antithetic Monte-Carlo and quasi Monte-Carlo methods as proposed in Philipson et al (2020).

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	if the model converged (conv=1 or 3), 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. If conv=2, V contains the second derivatives of the log-likelihood.
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,...). If `var.time` is specified, the corresponding measurement time is also included.
`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
`data`	the original data set (if returndata is TRUE)

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. doi:10.18637/jss.v078.i02

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 (2012). A universal approximate cross-validation criterion and its asymptotic distribution, Arxiv.

Philipson, Hickey, Crowther, Kolamunnage-Dona (2020). Faster Monte Carlo estimation of semi-parametric joint models of time-to-event and multivariate longitudinal data. Computational Statistics & Data Analysis 151.

Proust-Lima, Philipps, Perrot, Blanchin, Sebille (2021). Modeling repeated self-reported outcome data: a continuous-time longitudinal Item Response Theory model. https://arxiv.org/abs/2109.13064

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=0,X2=0,X3=0)
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).

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)
```

permut *Permutation of the latent classes*

Description

This function allows a re-ordering of the latent classes of an estimated model.

Usage

```
permut(m, order, estim = TRUE)
```

Arguments

m	an object inheriting from classes hlme, lcmm, multlcmm or Jointlcmm
order	a vector (integer between 1 and ng) containing the new order of the latent classes
estim	optional boolean specifying if the model should estimated with the reordered parameters as initial values. By default, the model is estimated. If FALSE, only the coefficients in $best are modified. All other outputs are not changed.

Value

An object of the same class as m, with reordered classes, or the initial object with new coefficients if estim is FALSE.

Author(s)

Viviane Philipps and Cecile Proust-Lima

Examples

```
## Estimation of a hlme model with 2 classes
m2 <- 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))

## Exchange class 2 and class 1
m2b <- permut(m2,order=c(2,1))
```

plot *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, shades, ...)

## S3 method for class 'lcmm'
plot(x, which = "residuals", var.time, break.times, marg, subset, shades, ...)

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

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

## S3 method for class 'mpjlcmm'
plot(x, which, event, ...)

## S3 method for class 'externSurv'
```

```
plot(x, which = "hazard", event, ...)

## S3 method for class 'externX'
plot(x, which = "postprob", event, ...)
```

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.
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.
shades	logical indicating if confidence intervals should be represented with shades. Default to FALSE, confidence intervals are represented as dotted lines.
...	other parameters to be passed through to plotting functions. This includes graphical parameters described in par function and further arguments 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.) and add (logical indicating if the curves should be added to an existing plot. Default to FALSE.).
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.

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",legend=c("linear","beta","3-Isplines","5-Isplines"),
col=1:4,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.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. does. ~~

Usage

```
## S3 method for class 'cuminc'
plot(
  x,
  profil = 1,
  event = 1,
  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.Diffepoce *Plots*

Description

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

Usage

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

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

Arguments

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

Details

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

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.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.ItemInfo` *Plot of information functions*

Description

This function plots the information functions stemmed from a `lcmm` or `multlcmm` object with ordinal outcomes modeled via threshold links.

Usage

```
## S3 method for class 'ItemInfo'
plot(
  x,
  which = "ItemInfo",
  outcome = "all",
  legend.loc = "topright",
  legend = NULL,
  add = FALSE,
  shades = TRUE,
  ...
)
```

Arguments

`x`	an object inheriting from classes `ItemInfo`
`which`	character specifying the values to plot. Should be one of 'ItemInfo' for the Fisher information function of the ordinal outcomes, 'LevelInfo' for the information of each item's level or 'LevelProb' for the probability of the item's levels. Default to 'ItemInfo'.
`outcome`	character specifying the outcome to consider. Default to "all".
`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)

Viviane Philipps and Cecile Proust-Lima

`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 'predictYcond'
plot(x, legend.loc = "topleft", legend, add = FALSE, shades = TRUE, ...)
```

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.
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
outcome	for predictY and multivariate model fitted with multlcmm only, the outcome to consider.

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, Jointlcmm, mpjlcmm, externSurv or externX object.

Usage

```
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)
```

---

predictClass *Posterior classification and class-membership probabilities*

Description

This function provides the posterior classification and posterior individual class-membership probabilities for external data.

Usage

```
predictClass(model, newdata, subject = NULL)
```

Arguments

model	an object inheriting from class hlme, lcmm, Jointlcmm or multlcmm representing a general latent class mixed model.
newdata	data frame containing the data from which predictions are to be computed. The data frame should include at least all the covariates listed in model$Xnames2, the outcome(s) and the grouping structure. Names should match exactly.
subject	character specifying the name of the grouping structure. If NULL (the default), the same as in the model will be used.

Value

a matrix with 2+ng columns: the grouping structure, the predicted class and the ng posterior class-membership probabilities.

Author(s)

Sasha Cuau, Viviane Philipps, Cecile Proust-Lima

Examples

```
## Not run:
library(NormPsy)
paquid$normMMSE <- normMMSE(paquid$MMSE)
paquid$age65 <- (paquid$age - 65)/10
m2b <- hlme(normMMSE ~ age65+I(age65^2)+CEP, random =~ age65+I(age65^2), subject = 'ID',
data = paquid, ng = 2, mixture =~ age65+I(age65^2), B = c(0, 60, 40, 0, -4, 0, -10, 10,
212.869397, -216.421323,456.229910, 55.713775, -145.715516, 59.351000, 10.072221))
predictClass(m2b, newdata=paquid[1:6,])

## End(Not run)
```

`predictCor` *Prediction of the Brownian motion or the autoregressive correlation*

Description

The function computes the predicted values of the BM or AR correlation given observed data provided in input.

Usage

```
predictCor(model, newdata, predtimes)
```

Arguments

`model`	an object inheriting from class `hlme`, `lcmm`, `Jointlcmm` or `multlcmm` representing a general latent class mixed model.
`newdata`	data frame containing the data from which predictions are to be computed. The data frame should include at least all the covariates listed in model$Xnames2, and the marker(s) values. Names should match exactly the names of the variables in the model.
`predtimes`	numeric vector containing the prediction times

Value

a matrix containing the predicted correlation in each latent class

Author(s)

Viviane Philipps, Cecile Proust-Lima

Examples

```
## Not run:
 m <- hlme(fixed = Y ~ Time, mixture = ~1, random = ~1, subject = "ID",
  ng = 2, cor = BM(Time), data = data_hlme, B = c(0,20,30,-1,5,2,0.1))
 predictCor(m, newdata = data_hlme[1:3, ], predtimes = seq(0, 5, 0.5))

## End(Not run)
```

`predictL` *Class-specific marginal or subject-specific 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`, `multlcmm` and `Jointlcmm` objects. The function `predictY` provides the class-specific predicted trajectories computed in the natural scale of the outcome(s).

Usage

```
predictL(
  x,
  newdata,
  var.time,
  na.action = 1,
  confint = FALSE,
  predRE = NULL,
  predCor = NULL,
  ...
)
```

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.
predRE	optional data frame containing the predicted random effects in each latent class. If NULL (the default), marginal prediction are computed.If predRE is specified, subject-specific predictions are computed.
predCor	optional matrix. If predCor is specified, the returned predictions include the correlation (BM or AR) part of the model. predCor should be a matrix with as many columns as latent classes and as many rows as newdata. In particular, predCor can be the result of a predictCor call.
...	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

predictY,predictRE,predictYcond

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

```
predictlink(x, ndraws, 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)
```

`predictRE` *Predictions of the random-effects*

Description

The function computes the predicted values of the random effects given observed data provided in input. With multiple latent classes, these predictions are averaged over classes using the posterior class-membership probabilities.

Usage

```
predictRE(model, newdata, subject = NULL, classpredRE = FALSE)
```

Arguments

`model`	an object inheriting from class `hlme`, `lcmm`, `Jointlcmm` or `multlcmm` representing a general latent class mixed model.
`newdata`	data frame containing the data from which predictions are to be computed. The data frame should include at least all the covariates listed in model$Xnames2, the marker(s) values and the grouping structure. Names should match exactly the names of the variables in the model.

subject	character specifying the name of the grouping structure. If NULL (the default), the same as in the model will be used.
classpredRE	logical indicating if class specific random effects should be returned. By default, classpredRE is FALSE, so that the random effects are aggregated over the classes.

Value

a data frame containing the grouping structure and the predicted random-effects.

Author(s)

Sasha Cuau, Viviane Philipps, Cecile Proust-Lima

Examples

```
## Not run:
library(NormPsy)
paquid$normMMSE <- normMMSE(paquid$MMSE)
paquid$age65 <- (paquid$age - 65)/10
m2b <- hlme(normMMSE ~ age65+I(age65^2)+CEP, random =~ age65+I(age65^2), subject = 'ID',
data = paquid, ng = 2, mixture =~ age65+I(age65^2), B = c(0, 60, 40, 0, -4, 0, -10, 10,
212.869397, -216.421323,456.229910, 55.713775, -145.715516, 59.351000, 10.072221))
predictRE(m2b,newdata=paquid[1:6,])

## End(Not run)
```

---

predictY

Predictions (marginal and possibly subject-specific in some cases) of a hlme, lcmm, multlcmm *or* Jointlcmm *object in the natural scale of the longitudinal outcome(s) computed from a profile of covariates.*

Description

For hlme and Jointlcmm objects, the function computes the predicted values of the longitudinal marker (in each latent class of ng>1) for a specified profile of covariates. Marginal or subject-specific predictions can be obtained. For lcmm and multlcmm objects, only marginal prediction are available, and 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 'Jointlcmm'
predictY(
  x,
  newdata,
  var.time,
  methInteg = 0,
  nsim = 20,
  draws = FALSE,
  ndraws = 2000,
  na.action = 1,
  predRE = NULL,
  predCor = NULL,
  ...
)

## S3 method for class 'hlme'
predictY(
  x,
  newdata,
  var.time,
  draws = FALSE,
  na.action = 1,
  predRE = NULL,
  predCor = NULL,
  ...
)

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

predictY(x, newdata, var.time, ...)

## 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 to be 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. For hlme object and marg=FALSE, the grouping structure and values for the outcome should also be specified.
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.
predRE	option only available for hlme or Jointlcmm models. If predRE is specified, the predictY function returns subject-specific predictions. If NULL, marginal predictions are computed. For subject-specific predictions, predRE should be a

data frame with as many rows as latent classes containing the predicted random effects in each latent class. In particular, predRE can be the result of a predictRE call with classpredRE = TRUE.

`predCor` option only available for `hlme` models and `Jointlcmm` models without any link function. If `predCor` is specified, the returned predictions include the correlation (BM or AR) part of the model. `predCor` should be a matrix with as many columns as latent classes and as many rows as newdata. In particular, predCor can be the result of a predictCor call.

`...` further arguments to be passed to or from other methods. Only the argument 'median' will be used, other are ignored. 'median' should be a logical indicating whether the median should be computed. By default, the mean value is computed.

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.

For objects of class `hlme` with `marg=FALSE`, returns a matrix with 2+ng columns : the grouping structure, subject-specific predictions (pred_ss) averaged over classes and the class-specific subject-specific predictions (with the number of the latent class: pred_ss_1,pred_ss_2,...)

- `times` : the `var.time` variable from `newdata`

Author(s)

Cecile Proust-Lima, Viviane Philipps, Sasha Cuau

See Also

`predictYback`

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)
```

`predictYback` *Marginal predictions in the natural scale of a pre-transformed outcome*

Description

The function computes the predicted values of the longitudinal marker (in each latent class if ng>1) for a specified profile of covariates, when a non-parameterized pre-transformation was applied (e.g., log, square root). A Gauss-Hermite or Monte-Carlo integration is used to numerically compute the back-transformed predictions.

Usage

```
predictYback(
  x,
  newdata,
  var.time,
```

```
  methInteg = 0,
  nsim = 20,
  draws = FALSE,
  ndraws = 2000,
  na.action = 1,
  back,
  ...
)
```

Arguments

x	an object inheriting from class hlme representing a general latent class mixed model.
newdata	data frame containing the data from which predictions are to be 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, i.e., the names of covariates specified in hlme 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. 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 accounts for the correlation between the predicted values.
nsim	number of points used in the numerical integration. 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	boolean specifying whether confidence bands should be computed. If draws=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	integer. 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.
back	function to back-transform the outcome in the original scale.
...	further arguments to be passed to or from other methods. They are ignored in this function.

Value

An object of class predictY.

Examples

```
data_lcmm$transfYdep2 <- sqrt(30 - data_lcmm$Ydep2)
m1 <- hlme(transfYdep2 ~ Time, random=~ Time, subject="ID", data = data_lcmm)
pred1 <- predictYback(m1, newdata = data.frame(Time = seq(0, 3, 0.1)),
var.time = "Time", back = function(x) {30 - x^2})
plot(pred1)
```

`predictYcond` *Conditional predictions of a* `lcmm`*,* `multlcmm` *or* `Jointlcmm` *object in the natural scale of the longitudinal outcome(s) for specified latent process values.*

Description

The function computes the predicted values of the longitudinal markers in their natural scale for specified values of the latent process. For splines and Beta links, a Gauss-Hermite integration is 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

```
predictYcond(
  x,
  lprocess,
  condRE_Y = FALSE,
  nsim = 200,
  draws = FALSE,
  ndraws = 2000,
  ...
)
```

Arguments

`x`	an object inheriting from class `lcmm`, `Jointlcmm` or `multlcmm` representing a general latent class mixed model.
`lprocess`	numeric vector containing the latent process values at which the predictions should be computed.
`condRE_Y`	for multlcmm objects only, logical indicating if the predictions are conditional to the outcome specific random effects or not. Default to FALSE, the predictions are marginal to these random effects.
`nsim`	number of points used in the numerical integration (Monte-Carlo) with splines or Beta link functions. nsim should be relatively important (nsim=200 by default).

draws	optional boolean specifying whether median and confidence bands of the predicted values should be computed (TRUE) - whatever the type of link function. 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.
...	further arguments to be passed to or from other methods. They are ignored in this function.

Value

An object of class predictYcond with values :

- pred : If draws=FALSE, returns a matrix with 3 columns : the first column indicates the name of the outcome, the second indicates the latent process value and the last is the computed prediction. If draws=TRUE, returns a matrix with 5 columns : the name of the outcome, the latent process value and the 50%, 2.5% and 97.5% percentiles of the approximated posterior distribution of predicted values.

- object : the model from which the predictions are computed.

Author(s)

Cecile Proust-Lima, Viviane Philipps

See Also

predictY, predictlink

Examples

```
## Not run:
m12 <- lcmm(Ydep2~Time+I(Time^2),random=~Time,subject='ID',ng=1,
data=data_lcmm,link="3-equi-splines")
predm12 <- predictYcond(m12,lprocess=seq(-8,2,length.out=100),draws=TRUE)
plot(predm12)

## 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 'lcmm'
print(x, ...)
```

Arguments

x	an object inheriting from classes hlme, lcmm, multlcmm for fitted latent class mixed-effects, or class Jointlcmm, mpjclmm for a Joint latent class mixed model or epoce for predictive accuracy computations or externSurv, externX for secondary regression models.
...	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

sampleParameters *Sample models parameters*

Description

Generates parameters from models estimated with hlme, lcmm, multlcmm, Jointlcmm or mpjlcmm functions. Parameters are randomly sampled from a Gaussian distribution using estimates and corresponding variances of a model.

Usage

```
sampleParameters(x, cholesky = FALSE)
```

Arguments

x	an object of class hlme, lcmm, multlcmm, Jointlcmm, or mpjlcmm.
cholesky	optional logical indicating if cholesky parameters should be returned. Default to FALSE, the variance-covariance parameters of the random effects are returned.

Value

a named vector containing the sampled parameters.

Author(s)

Viviane Philipps

Examples

```
m <- hlme(Y ~ Time * X1, random = ~ Time, subject = 'ID', data = data_hlme)

sampleParameters(m)
```

---

simdataHADS *Simulated dataset simdataHADS*

Description

The data mimic the PREDIALA study described and analyzed in Proust-Lima et al (2021 - https://arxiv.org/abs/2109.13064). The study aims to describe the trajectories of depressive symptomatology of patients suffering end-stage renal disease and registered on the renal transplant waiting list. Repeated measures of anxiety and depression (HADS) were simulated at different times of measurement for 561 subjects. Four time-independent covariates were also generated: group (dialyzed or pre-emptive), sex and age at entry in the cohort and time on the waiting list at entry in the cohort.

Format

A data frame with 1140 observations on the following 13 variables.

grp group with 0=dialyzed and 1=preemptive

sex sex with 0=woman and 1=man

age age at entry in the cohort

hads_2 item 2 of HADS measuring depression

hads_4 item 4 of HADS measuring depression

hads_6 item 6 of HADS measuring depression

hads_8 item 8 of HADS measuring depression

hads_10 item 10 of HADS measuring depression

hads_12 item 12 of HADS measuring depression

hads_14 item 14 of HADS measuring depression

ID subject identification number

time time of measurement

time_entry time on the waiting list at entry in the cohort

`simulate.lcmm` *Data simulation according to models from lcmm package*

Description

This function simulates a sample according to a model estimated with `hlme`, `lcmm`, `multlcmm` or `Jointlcmm` functions.

Usage

```
## S3 method for class 'lcmm'
simulate(
  object,
  nsim,
  seed,
  times,
  tname = NULL,
  n,
  Xbin = NULL,
  Xcont = NULL,
  entry = 0,
  dropout = NULL,
  pMCAR = 0,
  ...
)
```

Arguments

`object`	an object of class `hlme`, `lcmm`, `multlcmm` or `Jointlcmm`
`nsim`	not used (for compatibility with stats::simulate). The function simulates only one sample
`seed`	the random seed
`times`	either a data frame with 2 columns containing IDs and measurement times, or a vector of length 4 specifying the minimal and maximum measurement times, the spacing between 2 consecutive visits and the margin around this spacing
`tname`	the name of the variable representing the measurement times in `object`. Default to the second column's name of times if it is a data frame, and to object$var.time otherwise.
`n`	number of subjects to simulate. Required only if times is not a data frame.

Xbin	an optional named list giving the probabilities of the binary covariates to simulate. The list's names should match the binary covariate's names used in object.
Xcont	an optional named list giving the mean and standard deviation of the Gaussian covariates to simulate. The list's names should match the continuous covariate's names used in object.
entry	expression to simulate a subject's entry time. Default to 0.
dropout	expression to simulate a subject's time to dropout. Default to NULL, no dropout is considered.
pMCAR	optional numeric giving an observation's probability to be missing. Default to 0, no missing data are introduced.
...	additionnal options. None is used yet.

Value

a data frame with one line per observation and one column per variable. Variables appears in the following order : subject id, measurement time, entry time, binary covariates, continuous covariates, longitudinal outcomes, latent class, entry time, survival time, event indicator.

Author(s)

Viviane Philipps and Cecile Proust-Lima

Examples

```
## estimation of a 2 classes mixed model
m2 <- 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))

## simulate according to model m2 with same number of subjects and
## same measurement times as in data_lcmm. Binary covariates X1 and X2 are simulated
## according to a Bernoulli distribution with probability p=0.5, continuous covariate
## X3 is simulated according to a Gaussian distribution with mean=1 and sd=1 :
dsim1 <- simulate(m2, times=data_hlme[,c("ID","Time")],
                  Xbin=list(X1=0.5, X2=0.5), Xcont=list(X3=c(1,1)))

## simulate a dataset of 300 subjects according to the same model
## with new observation times, equally spaced and ranging from 0 to 3 :
dsim2 <- simulate(m2, times=c(0,3,0.5,0), n=300, tname="Time",
                  Xbin=list(X1=0.5, X2=0.5), Xcont=list(X3=c(1,1)))
```

StandardMethods *Standard methods for estimated models*

Description

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

Usage

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

summary.lcmm *Summary of a* hlme, lcmm, Jointlcmm, multlcmm, mpjlcmm, externSurv, externX 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 'lcmm'
summary(object, ...)
```

Arguments

object	an object inheriting from classes hlme, lcmm, multlcmm for fitted latent class mixed-effects, or class Jointlcmm, mpjlcmm for a Joint latent class mixed model or epoce or Diffepoce for predictive accuracy computations or externSurv, externX for secondary regression models.
...	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

summaryplot *Summary of models*

Description

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

Usage

```
summaryplot(
  m1,
  ...,
  which = c("BIC", "entropy", "ICL"),
  mfrow = c(1, length(which)),
  xaxis = "G"
)
```

Arguments

m1	an object of class hlme, lcmm, multlcmm, Jointlcmm, mpjlcmm, externVar or externVar.
...	further arguments, in particular other objects of class hlme, lcmm, multlcmm, Jointlcmm or mpjlcmm, and graphical parameters.
which	character vector indicating which results should be plotted. Possible values are "loglik", "conv", "npm", "AIC", "BIC", "SABIC", "entropy", "ICL", "ICL1", "ICL2".
mfrow	for multiple plots, number of rows and columns to split the graphical device. Default to one line and length(which) columns.
xaxis	the abscissa of the plot. Default to "G", the number of latent classes.

Details

Can be reported the usual criteria used to assess the fit and the clustering of the data: - maximum log-likelihood L (the higher the better) - number of parameters P, number of classes G, convergence criterion (1 = converged) - AIC (the lower the better) computed as -2L+2P - BIC (the lower the better) computed as -2L+ P log(N) where N is the number of subjects - SABIC (the lower the better) computed as -2L+ P log((N+2)/24) - Entropy (the closer to one the better) computed as 1+sum[pi_ig*log(pi_ig)]/(N*log(G)) where pi_ig is the posterior probability that subject i belongs to class g - ICL (the lower the better) computed in two ways : ICL1 = BIC - sum[pi_ig*log(pi_ig)] or ICL2 = BIC - 2*sum(log(max(pi_ig)), where the max is taken over the classes for each subject. - %class computed as the proportion of each class based on c_ig

Author(s)

Sasha Cuau, Viviane Philipps, Cecile Proust-Lima

See Also

summary, summarytable

Examples

```
## Not run:
library(NormPsy)
paquid$normMMSE <- normMMSE(paquid$MMSE)
paquid$age65 <- (paquid$age - 65)/10
m1 <- hlme(normMMSE~age65+I(age65^2)+CEP, random=~age65+I(age65^2), subject='ID', data=paquid)
m2 <- hlme(normMMSE~age65+I(age65^2)+CEP, random=~age65+I(age65^2), subject='ID', data=paquid,
ng = 2, mixture=~age65+I(age65^2), B=m1)
m3g <- gridsearch(hlme(normMMSE~age65+I(age65^2)+CEP, random=~age65+I(age65^2), subject='ID',
data=paquid, ng=3, mixture=~age65+I(age65^2)), rep=100, maxiter=30, minit=m1)
summaryplot(m1, m2, m3g, which=c("BIC","entropy","ICL"),bty="l",pch=20,col=2)

## End(Not run)
```

---

summarytable *Summary of models*

Description

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

Usage

```
summarytable(
  m1,
  ...,
  which = c("G", "loglik", "npm", "BIC", "%class"),
  display = TRUE
)
```

Arguments

m1	an object of class hlme, lcmm, multlcmm, Jointlcmm, mpjlcmm, externVar or externVar.
...	further arguments, in particular other objects of class hlme, lcmm, multlcmm, Jointlcmm or mpjlcmm.
which	character vector indicating which results should be returned. Possible values are "G", "loglik", "conv", "npm", "AIC", "BIC", "SABIC", "entropy", "ICL", "ICL1", "ICL2", "%class".
display	logical indicating whether the table should be printed (the default) or not (display=FALSE)

Details

Can be reported the usual criteria used to assess the fit and the clustering of the data: - maximum log-likelihood L (the higher the better) - number of parameters P, number of classes G, convergence criterion (1 = converged) - AIC (the lower the better) computed as -2L+2P - BIC (the lower the better) computed as -2L+ P log(N) where N is the number of subjects - SABIC (the lower the better) computed as -2L+ P log((N+2)/24) - Entropy (the closer to one the better) computed as 1+sum[pi_ig*log(pi_ig)]/(N*log(G)) where pi_ig is the posterior probability that subject i belongs to class g - ICL (the lower the better) computed in two ways : ICL1 = BIC - sum[pi_ig*log(pi_ig)] or ICL2 = BIC - 2*sum(log(max(pi_ig)), where the max is taken over the classes for each subject. - %class computed as the proportion of each class based on c_ig

Value

a matrix giving for each model the values of the requested indexes. By default, 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

`update.mpjlcmm` *Update the longitudinal submodels*

Description

This function updates the longitudinal submodels of a mpjlcmm object by injecting the estimated parameters and their variances in each hlme/lcmm/multlcmm model used to define the multi-process joint model. The same (uni-process) models as specified in the mpjlcmm call are returned, with updated outputs for best, V, conv, cholesky, pred, predRE, predRE_Y, pprob. All postfit functions (plots, predictions, ...) can then be called on the updated hlme/lcmm/multlcmm models. See mpjlcmm's help page for examples.

Usage

```
## S3 method for class 'mpjlcmm'
update(object, ...)
```

Arguments

`object`	an estimated mpjlcmm model
`...`	not used

Value

A list of hlme/lcmm/multlcmm models. The models appear in the same order as specified in the call to the mpjlcmm function.

`VarCov` *Variance-covariance of the estimates*

Description

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

Usage

```
VarCov(x)
```

Arguments

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

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`

`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

```
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

```
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, Jointlcmm or mpjlcmm models.

Usage

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

Arguments

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

`xclass` *Cross classifications*

Description

This function crosses the posterior classifications of two estimated models

Usage

```
xclass(m1, m2)
```

Arguments

`m1`	an object inheriting from classes `hlme`, `lcmm`, `multlcmm`, `Jointlcmm`, `mpjlcmm` or `externVar`
`m2`	an object inheriting from classes `hlme`, `lcmm`, `multlcmm`, `Jointlcmm`, `mpjlcmm` or `externVar`

Value

the contingency table of the two classifications

Author(s)

Viviane Philipps and Cecile Proust-Lima

Examples

```
## Estimation of the models
m2 <- 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))
m3 <- hlme(fixed = Y ~ Time * X1, mixture = ~Time, random = ~Time,subject = "ID",
classmb = ~X2 + X3, ng = 3, data = data_hlme,B=c(-0.21, 0.31, -2.11, -0.81, -0.24,
-0.18, 25.4, 20.09, 30.18, -0.43, -1.1, 0.25, 2.37, -0.29, 2.34, 0.03, 0.74, 0.97))

## Compare the classifications
xclass(m2,m3)
# The 39 subjects in class 2 of m3 come from class 1 of m2.
# In the same way, all the subjects in class 3 come from class 2 of m2.
# Class 1 of m3 mixes subject from class 1 and class 2 of m2.
```

Index

∗ **datasets**
  data_hlme, 7
  data_lcmm, 8
  paquid, 74
  simdataHADS, 103
∗ **print**
  print.lcmm, 102
  summary.lcmm, 107
.Contlcmm *(*ForInternalUse*)*, 24
.Ordlcmm *(*ForInternalUse*)*, 24
.plotbaselinerisk *(*ForInternalUse*)*, 24
.plotfit *(*ForInternalUse*)*, 24
.plotlinkfunctionmult *(*ForInternalUse*)*, 24
.plotlinkfuntion *(*ForInternalUse*)*, 24
.plotpostprob *(*ForInternalUse*)*, 24
.plotresid *(*ForInternalUse*)*, 24
.plotsurvival *(*ForInternalUse*)*, 24

Brandom *(*ForInternalUse*)*, 24

C_calculustransfo *(*ForInternalUse*)*, 24
C_cvpl *(*ForInternalUse*)*, 24
C_hetmixcont *(*ForInternalUse*)*, 24
C_hetmixcontmult *(*ForInternalUse*)*, 24
C_hetmixlin *(*ForInternalUse*)*, 24
C_hetmixord *(*ForInternalUse*)*, 24
C_jointhet *(*ForInternalUse*)*, 24
C_postprob2 *(*ForInternalUse*)*, 24
C_predictcont *(*ForInternalUse*)*, 24
C_predictmult *(*ForInternalUse*)*, 24
coef.externSurv *(*StandardMethods*)*, 106
coef.externX *(*StandardMethods*)*, 106
coef.hlme *(*StandardMethods*)*, 106
coef.Jointlcmm *(*StandardMethods*)*, 106
coef.lcmm *(*StandardMethods*)*, 106
coef.mpjlcmm *(*StandardMethods*)*, 106
coef.multlcmm *(*StandardMethods*)*, 106
createVarianceMatrix, 5
cuminc, 6, *81*

data_hlme, 7
data_lcmm, 8
Diffepoce, 9, *82*
dynpred, 11, *83*

epoce, *10*, 13, *42*, *82*
estimates, 15, *111*
externVar, 16

factor.names *(*ForInternalUse*)*, 24
fitted.hlme *(*StandardMethods*)*, 106
fitted.Jointlcmm *(*StandardMethods*)*, 106
fitted.lcmm *(*StandardMethods*)*, 106
fitted.multlcmm *(*StandardMethods*)*, 106
fitY, 23
fixef.hlme *(*StandardMethods*)*, 106
fixef.Jointlcmm *(*StandardMethods*)*, 106
fixef.lcmm *(*StandardMethods*)*, 106
fixef.multlcmm *(*StandardMethods*)*, 106
ForInternalUse, 24

gridsearch, 25

hlme, *7, 8*, *16*, 26, *51*, *78*, *86*, *88*, *102*, *107*, *110*, *113*

ItemInfo, 32

jlcmm *(*Jointlcmm*)*, 34
Jointlcmm, *7, 8*, *10*, *12*, *15, 16*, 34, *78*, *81*, *86*, *88*, *102*, *107*, *110*, *113*

lcmm, *8*, *16*, 44, *73*, *78*, *86*, *88*, *92*, *102*, *107*, *110*, *113*
lcmm-package, 3
loglik, 53
loglikhlme *(*loglik*)*, 53
loglikJointlcmm *(*loglik*)*, 53
logliklcmm *(*loglik*)*, 53
loglikmpjlcmm *(*loglik*)*, 53
loglikmultlcmm *(*loglik*)*, 53

mixture (ForInternalUse), 24
mlcmm (multlcmm), 65
mpjlcmm, 60
multlcmm, *16*, 65, *78*, *86*, *92*, *107*, *110*, *113*

paquid, 74
permut, 75
plot, 76
plot.cuminc, *7*, 80
plot.Diffepoce, 81
plot.dynpred, *12*, 82
plot.epoce, *15*
plot.epoce (plot.Diffepoce), 81
plot.hlme, *31*
plot.ItemInfo, 84
plot.Jointlcmm, *7*, *42*, *81*
plot.lcmm, *24*, *51*, *88*, *92*
plot.multlcmm, *73*
plot.pred.accuracy (plot.Diffepoce), 81
plot.predict, *7*, *12*, *42*, *51*, 85
plot.predictL (plot.predict), 85
plot.predictlink, *92*
plot.predictlink (plot.predict), 85
plot.predictY (plot.predict), 85
plot.predictYcond (plot.predict), 85
postprob, *7*, *31*, *42*, *51*, *73*, 87
predictClass, 88
predictCor, 89
predictL, *73*, 90
predictlink, 92, *101*
predictRE, *91*, 93
predictY, *12*, *24*, *31*, *73*, *91*, 94, *101*
predictYback, *97*, 98
predictYcond, *91*, 100
print.Diffepoce (print.lcmm), 102
print.epoce, *15*
print.epoce (print.lcmm), 102
print.externSurv (print.lcmm), 102
print.externX (print.lcmm), 102
print.hlme (print.lcmm), 102
print.Jointlcmm (print.lcmm), 102
print.lcmm, 102
print.mpjlcmm (print.lcmm), 102
print.multlcmm (print.lcmm), 102

ranef.hlme (StandardMethods), 106
ranef.Jointlcmm (StandardMethods), 106
ranef.lcmm (StandardMethods), 106
ranef.multlcmm (StandardMethods), 106
residuals.hlme (StandardMethods), 106
residuals.Jointlcmm (StandardMethods), 106
residuals.lcmm (StandardMethods), 106
residuals.multlcmm (StandardMethods), 106
risq_spl (ForInternalUse), 24
risqcum_spl (ForInternalUse), 24

sampleParameters, 102
simdataHADS, 103
simulate.lcmm, 104
StandardMethods, 106
summary, *31*, *109*, *110*
summary.Diffepoce, *10*
summary.Diffepoce (summary.lcmm), 107
summary.epoce, *15*
summary.epoce (summary.lcmm), 107
summary.externSurv (summary.lcmm), 107
summary.externX (summary.lcmm), 107
summary.hlme (summary.lcmm), 107
summary.Jointlcmm (summary.lcmm), 107
summary.lcmm, *7*, 107
summary.mpjlcmm (summary.lcmm), 107
summary.multlcmm (summary.lcmm), 107
summaryplot, 108
summarytable, 109, *109*

update.mpjlcmm, 110

VarCov, *16*, 111
VarCovRE, 111
VarExpl, 112
vcov.externSurv (StandardMethods), 106
vcov.externX (StandardMethods), 106
vcov.hlme (StandardMethods), 106
vcov.Jointlcmm (StandardMethods), 106
vcov.lcmm (StandardMethods), 106
vcov.mpjlcmm (StandardMethods), 106
vcov.multlcmm (StandardMethods), 106

WaldMult, 113

xclass, 114