Package 'JSM'

June 9, 2025

Type Package

Title Semiparametric Joint Modeling of Survival and Longitudinal Data

Version 1.0.2

Date 2025-06-09

Maintainer Cong Xu <helenxu1112@gmail.com>

Description
Maximum likelihood estimation for the semiparametric joint modeling of survival and longitudinal data. Refer to the Journal of Statistical Software article: <doi:10.18637/jss.v093.i02>.

License BSD_3_clause + file LICENSE

Imports Rcpp (>= 0.11.5)

LinkingTo Rcpp, RcppEigen

NeedsCompilation yes

Depends R (>= 3.0.0), nlme, splines, statmod, survival

Suggests testthat

LazyData true

Repository CRAN

Date/Publication 2025-06-09 12:40:02 UTC

Author Cong Xu [aut, cre],
Pantelis Z. Hadjipantelis [aut],
Jane-Ling Wang [aut]

Contents

`aids` *ddI versus ddC in HIV-infected Patients*

Description

A randomized clinical trial in which both survival and longitudinal data were collected to compare the efficacy and safety of two antiretroviral drugs, namely `ddI` (didanosine) and `ddC` (zalcitabine), in treating HIV-infected patients intolerant or failing zidovudine (AZT) therapy.

Format

A data frame with 1405 observations on the following 12 variables.

`ID` patient ID, there are 467 patients in total.

`Time` survival time, i.e. time to death or censoring.

`death` death indicator: 0 denotes censoring; 1 denotes death.

`obstime` time points at which the longitudinal measurements, i.e. CD4 cell counts, are recorded.

`CD4` CD4 cell counts measured at `obstime`.

`drug` drug indicator with two levels: `ddI` and `ddC`.

`gender` gender indicator with two levels: `male` and `female`.

`prevDiag` AIDS diagnosis at study entry indicator with two levels: `AIDS` and `noAIDS`.

`AZT` AZT intolerance/failure indicator with two levels: `intolerance` and `failure`.

`start` same with `obstime`, starting time of the interval which contains the time of the CD4 cell count measurement.

`stop` ending time of the interval which contains the time of the CD4 cell count measurement.

`event` event indicator suggesting whether the event-of-interest, i.e. death, happens in the interval given by `start` and `stop`.

Source

Goldman, A., Carlin, B., Crane, L., Launer, C., Korvick, J., Deyton, L. and Abrams, D. (1996) Response of CD4+ and clinical consequences to treatment using ddI or ddC in patients with advanced HIV infection. *Journal of Acquired Immune Deficiency Syndromes and Human Retrovirology* **11**, 161–169.

References

Guo, X. and Carlin, B. (2004) Separate and joint modeling of longitudinal and event time data using standard computer packages. *The American Statistician* **58**, 16–24.

Xu, C., Baines, P. D. and Wang, J. L. (2014) Standard error estimation using the EM algorithm for the joint modeling of survival and longitudinal data. *Biostatistics* **15**, 731–744

Examples

```
head(aids)
```

`confint` *Obtain Confidence Intervals for Joint Model Parameters*

Description

`confint` is a generic function which computes confidence intervals for parameters in models fitted by `jmodelTM()` or `jmodelMult()`.

Usage

```
## S3 method for class 'jmodelTM'
confint(object, parm, level = 0.95, ...)
## S3 method for class 'jmodelMult'
confint(object, parm, level = 0.95, ...)
```

Arguments

`object`	an object inheriting from class `jmodelTM` or `jmodelMult`.
`parm`	a specification of which parameters are to be given confidence intervals. As currently implemented, always give confidence intervals for all regression coefficients.
`level`	the confidence level required.
`...`	additional arguments required. None is used in this method.

Value

A list consists of the following components:

`infoLong`	a matrix with columns giving parameter estimates as well as their lower and upper confidence limits for the regression parameters of the longitudinal process.
`infoSurv`	a matrix with columns giving parameter estimates as well as their lower and upper confidence limits for the regression parameters of the survival process.
`level`	the confidence level used in computing the confidence limits.

Author(s)

Cong Xu <helenxu1112@gmail.com>

Examples

```
## Not run:
fitLME <- lme(proth ~ Trt * obstime, random = ~ 1 | ID, data = liver)
fitCOX <- coxph(Surv(start, stop, event) ~ Trt, data = liver, x = TRUE)
fitJT.ph <- jmodelTM(fitLME, fitCOX, liver, timeVarY = 'obstime')

# 95% confidence intervals for the joint model parameters
confint(fitJT.ph)

## End(Not run)
```

`dataPreprocess` *Preprocess Data to Be Fed into Joint Models*

Description

`dataPreprocess` is a function to preprocess data to be used in fitting joint models. Suppose the situation is that the longitudinal measurements are recorded in a data frame with one row per measurment and the survival information are recorded in another data frame with one row per subject. This function merges the two data frames by subject identification and generate three new columns: `start`, `stop`, `event`. See **Value**.

Usage

```
dataPreprocess(long, surv, id.col, long.time.col, surv.time.col, surv.event.col,
           surv.event.indicator = list(censored = 0, event = 1), suffix = ".join")
```

Arguments

`long`	a data frame for the longitudinal data, one row per measurment, with subject identification, time of measurement, and longitudinal measurements, etc.
`surv`	a data frame for the survival data, one row per subject, with subject identification (column name should match that in `long`), possibly censored time-to-event, and event indicator (normally 0=censored, 1=event), etc.
`id.col`	a `character` string specifying the subject identification column in both `long` and `surv`.
`long.time.col`	a `character` string specifying the time of measurement column in `long`.
`surv.time.col`	a `character` string specifying the possibly censored time-to-event column in `surv`.
`surv.event.col`	a `character` string specifying the event status column in `surv`.
`surv.event.indicator`	a `list` specifying the values in column `surv.event.col` corresponding to censored and event status.
`suffix`	a optional `character` string specifying the suffix to be added to the `start`, `stop`, `event` columns in case `long` or `surv` already have columns with these names.

Value

A data frame merging long and surv by subject identification, with one row per longitudinal measurment, and generate three new columns: start, stop, event (column names are added with suffix specified by suffix:

start starting time of the interval which contains the time of the longitudinal measurements.

stop ending time of the interval which contains the time of the longitudinal measurements.

event event indicator suggesting whether the event-of-interest, e.g. death, happens in the interval given by start and stop.

Note

1. If long and surv have columns sharing the same column names, the columns from long and surv would be named with suffixes ".long" and ".surv", respectively, in the output data frame. 2. The time of measurement of the longitudinal measurements and possibly censored time-to-event should be recorded consistently for each subject, i.e. time 0 means the same time point for the longitudinal and survival measurements.

Author(s)

Cong Xu <helenxu1112@gmail.com>

Examples

```
## Not run:
liver.join <- dataPreprocess(liver.long, liver.surv, 'ID', 'obstime', 'Time', 'death')

## End(Not run)
```

---

epilepsy *CBZ versus LTG in Epilepsy Patients*

Description

A randomised control trial, the SANAD (standard and new antiepileptic drugs) study, in which both survival and longitudinal data were collected to investigate the effect of drug titration on the relative effects of two antiepileptic drugs, namely CBZ (carbamazepine, a standard drug) and LTG (lamotrigine, a new drug), on treatment failure. Treatment failure, i.e. withdrawal of the randomized drug, is the event of interest. Two main reasons for withdrawal are unacceptable adverse effects (UAE) and inadequate seizure control (ISC).

Format

A data frame with 2797 observations on the following 16 variables.

ID patient ID, there are 605 patients in total.

Time survival time, i.e. time to withdrawal or censoring.

`withdraw` withdrawal indicator: 0 denotes censoring; 1 denotes withdrawal.

`withdrawUAE` withdrawal due to UAE indicator: 1 denotes withdrawal due to UAE; 0 otherwise.

`withdrawISC` withdrawal due to ISC indicator: 1 denotes withdrawal due to ISC; 0 otherwise.

`obstime` time points at which the longitudinal measurements, i.e. the dose, are recorded.

`dose` calibrated dose measured at `obstime`.

`drug` drug indicator with two levels: `CBZ` and `LTG`.

`age` age of patient at study entry.

`gender` gender indicator with two levels: `male` and `female`.

`disab` learning disability indicator with two levels: `No` and `Yes`.

`start` same with `obstime`, starting time of the interval which contains the time of the dose measurement.

`stop` ending time of the interval which contains the time of the dose measurement.

`event` event indicator suggesting whether the event-of-interest, i.e. withdrawal, happens in the interval given by `start` and `stop`.

`eventUAE` event indicator suggesting whether the event-of-interest, i.e. withdrawal due to UAE, happens in the interval given by `start` and `stop`.

`eventISC` event indicator suggesting whether the event-of-interest, i.e. withdrawal due to ISC, happens in the interval given by `start` and `stop`.

Source

Marson, A. G., AI-Kharusi, A. M., Alwaidh, M., Appleton, R., Baker, G. A., Chadwick, D. W., Cramp, C., Cockerell, O. C. Cooper, P. N., Doughty, J. et al. (2007) The SANAD study of effectiveness of carbamazepine, gabapentin, lamotrigine, oxcarbazepine, or topiramate for treatment of partial epilepsy: an unblinded randomised controlled trial. *The Lancet* **369**, 1000–1015.

References

Williamson P. R., Kolamunnage-Dona R., Philipson P. and Marson A. G. (2008) Joint modelling of longitudinal and competing risks data. *Statistics in Medicine* **27**, 6426–6438.

Examples

```
head(epilepsy)
```

`fitted` *Extract Fitted Values for Joint Models*

Description

`fitted` is a generic function which extracts fitted values from objects returned by `jmodelTM()` or `jmodelMult()`.

Usage

```
## S3 method for class 'jmodelTM'
fitted(object, process = c("Longitudinal", "Survival"),
        type = c("Marginal", "Conditional"), ...)
## S3 method for class 'jmodelMult'
fitted(object, process = c("Longitudinal", "Survival"),
        type = c("Marginal", "Conditional"), ...)
```

Arguments

object	an object inheriting from class jmodelTM or jmodelMult.
process	for which process the fitted values are calculated, i.e. the longitudinal or the survival process.
type	what type of fitted values to calculate for each process. See **Details**.
...	additional arguments required. None is used in this method.

Details

We have implemented the fitted value calculation for process = "Longitudinal" but not for process = "Survival" yet as they are not well defined under the joint modeling setting. There are two types of fitted values depending on whether to compute the values conditional on the random effects. With type = "Marginal", the fitted values are $\mathbf{X}_i^\top(t)\boldsymbol{\beta}$ for objects returned by jmodelTM() and $\mathbf{B}^\top(t)\boldsymbol{\gamma}$ for objects returned by jmodelMult(). With type = "Conditional", the fitted values are $\mathbf{X}_i^\top(t)\boldsymbol{\beta} + \mathbf{Z}_i^\top(t)\mathbf{b}_i$ for objects returned by jmodelTM() and $b_i \times \mathbf{B}^\top(t)\boldsymbol{\gamma}$ for objects returned by jmodelMult().

Value

A numeric vector of fitted values.

Author(s)

Cong Xu <helenxu1112@gmail.com>

Examples

```
## Not run:
fitLME <- lme(proth ~ Trt * obstime, random = ~ 1 | ID, data = liver)
fitCOX <- coxph(Surv(start, stop, event) ~ Trt, data = liver, x = TRUE)
fitJT.ph <- jmodelTM(fitLME, fitCOX, liver, timeVarY = 'obstime')

# fitted values for the longitudinal process
fitted(fitJT.ph, type = "Conditional")

## End(Not run)
```

`jmodelMult` *Semiparametric Joint Models for Survival and Longitudinal Data with Nonparametric Multiplicative Random Effects*

Description

This function applies a maximum likelihood approach to fit the semiparametric joint models of survival and normal longitudinal data. The survival model is assumed to come from a class of transformation models, including the Cox proportional hazards model and the proportional odds model as special cases. The longitudinal process is modeled by nonparametric multiplicative random effects (NMRE) model.

Usage

```
jmodelMult(fitLME, fitCOX, data, model = 1, rho = 0, timeVarY = NULL,
           timeVarT = NULL, control = list(), ...)
```

Arguments

`fitLME`	an object inheriting from class `lme` representing a fitted nonparametric multiplicative random effects model. See **Details** and **Note** and **Examples**.
`fitCOX`	an object inheriting from class `coxph` representing a fitted Cox proportional hazards regression model. Specifying `x = TRUE` is required in the call to `coxph()` to include the design matrix in the object fit. See **Note**.
`data`	a data.frame containing all the variables included in the joint modeling. See **Note**.
`model`	an indicator specifying the dependency between the survival and longitudinal outcomes. Default is 1. See **Details**.
`rho`	a nonnegative real number specifying the transformation model you would like to fit. Default is 0, i.e. the Cox proportional hazards model. See **Details**.
`timeVarY`	a character string indicating the time variable in the NMRE model. See **Examples**.
`timeVarT`	a character string indicating the time variable in the `coxph` object. Normally it is `NULL`. See **Note** and **Examples**.
`control`	a list of control values for the estimation algorithm with components: **tol.P** tolerance value for convergence in the parameters with default value 1e-03. See **Details**. **tol.L** tolerance value for convergence in the log-likelihood with default value 1e-06. See **Details**. **max.iter** the maximum number of EM iterations with default value 250. **SE.method** a character string specifying the standard error estimation method. Default is `"PRES"`. See **Details** and **Note**. **delta** a positive value used for numerical differentiation in the `SE.method`. Default is 1e-05 if `"PRES"` is used and 1e-03 otherwise. See **Details**.

nknot the number of Gauss-Hermite quadrature knots used to approximate the integrals over the random effects. Under the nonparametric multiplicative random effects model, there are only one-dimensional integrations and the default for `nknot` is 11.

... additional options to be passed to the `control` argument.

Details

The `jmodelMult` function fits joint models for survival and longitudinal data. Nonparametric multiplicative random effects models (NMRE) are assumed for the longitudinal processes. With the Cox proportional hazards model and the proportional odds model as special cases, a general class of transformation models are assumed for the survival processes. The baseline hazard functions are left unspecified, i.e. no parametric forms are assumed, thus leading to semiparametric models. For detailed model formulation, please refer to Xu, Hadjipantelis and Wang (2017).

The longitudinal model (NMRE) is written as

$$Y_i(t) = \mu_i(t) + \varepsilon_i(t) = b_i \times \mathbf{B}^\top(t)\boldsymbol{\gamma} + \varepsilon_i(t),$$

where $\mathbf{B}(t) = (B_1(t), \cdots, B_L(t))$ is a vector of B-spline basis functions and b_i is a random effect $\sim \mathcal{N}(1, \sigma_b^2)$. Note that we also allow the inclusion of baseline covariates as columns of $\mathbf{B}(t)$. If `model = 1`, then the linear predictor for the survival model is expressed as

$$\eta(t) = \mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha\mu_i(t),$$

indicating that the entire longitudinal process (free of error) enters the survival model as a covariate. If other values are assigned to the `model` argument, the linear predictor for the surival model is then expressed as

$$\eta(t) = \mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha b_i,$$

suggesting that the survival and longitudinal models only share the same random effect.

The survival model is written as

$$\Lambda(t|\eta(t)) = G\left[\int_0^t \exp\{\eta(s)\} d\Lambda_0(s)\right],$$

where $G(x) = \log(1+\rho x)/\rho$ with $\rho \geq 0$ is the class of logarithmic transfomrations. If `rho = 0`, then $G(x) = x$, yielding the Cox proportional hazards model. If `rho = 1`, then $G(x) = \log(1+x)$, yielding the proportional odds model. Users could assign any nonnegative real value to `rho`.

An expectation-maximization (EM) algorithm is implemented to obtain parameter estimates. The convergence criterion is either of (i) $\max\{|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}|/(|\boldsymbol{\theta}^{(t-1)}| + .Machine\$double.eps \times 2)\} < tol.P$, or (ii) $|L(\boldsymbol{\theta}^{(t)}) - L(\boldsymbol{\theta}^{(t-1)})|/(|L(\theta^{(t-1)})| + .Machine\$double.eps \times 2) < tol.L$, is satisfied. Here $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t-1)}$ are the vector of parameter estimates at the t-th and $(t-1)$-th EM iterations, respectively; $L(\boldsymbol{\theta})$ is the value of the log-likelihood function evaluated at $\boldsymbol{\theta}$. Users could specify the tolerance values `tol.P` and `tol.L` through the `control` argument.

For standard error estimation for the parameter estimates, three methods are provided, namely `"PRES"`, `"PFDS"` and `"PLFD"` (detailed information are referred to Xu, Baines and Wang (2014)). In the `control` argument, if `SE.method = "PRES"`, numerically differentiating the profile Fisher score vector with Richardson extrapolation is applied; if `SE.method = "PFDS"`, numerically differentiating the profile Fisher score vector with forward difference is applied; if `SE.method = "PLFD"`,

numerially (second) differentiating the profile likelihood with forward difference is applied. Generally, numerically differentiating a function $f(x)$ (an arbitrary function) with forward difference is expressed as

$$f'(x) = \frac{f(x+\delta) - f(x)}{\delta},$$

and that with Richardson extrapolation is expressed as

$$f'(x) = \frac{f(x-2\delta) - 8f(x-\delta) + 8f(x+\delta) - f(x+2\delta)}{12\delta}.$$

Users could specify the value of δ through the `delta` item in the `control` argument.

Value

See `jmodelMultObject` for the components of the fit.

Note

1. To fit a nonparametric multiplicative random effects model, the fixed effect in the `fitLME` object should be a matrix of B-spline basis functions (an object from the `bs` function) with the possibility of including baseline covariates and the random effect should only include a random intercept. In the `bs` function, it is a good practice to specify the boundary knots through the `Boundary.knots` argument, where the upper boundary knot is typically the longest follow-up time among all subjects. See **Examples**.

2. Currently, `jmodelMult()` could only handle the `fitLME` object with a simple random-effects structure (only the `pdDiag()` class). Moreover, the within-group correlation and heteroscedasticity structures in the `fitLME` object (i.e. the `correlation` and `weights` argument of `lme()`) are ignored.

3. The `data` argument in `jmodelMult()`, `lme()` and `coxph()` should be the same data frame.

4. For the `fitCOX` object, only the $W_i(t)$ in the linear predictor $\eta(t)$ for the survial model (see **Details**) should be involved in the `formula` argument of `coxph{}`. Since `coxph()` uses the same data frame as `lme()` does, a time-dependent Cox model must be fitted by `coxph()` although $W_i(t)$ may only contain time-independent covariates. See **Examples**.

5. If $W_i(t)$ in the linear predictor $\eta(t)$ for the survial model (see **Details**) does involve time-dependent covariate, then `timeVarT` must specify the name of the time variable involved (see **Examples**).

6. The standard error estimates are obtained by numerical approximations which is naturally subject to numerical errors. Therefore, in extreme cases, there may be `NA` values for part of the standard error estimates.

Author(s)

Cong Xu <helenxu1112@gmail.com> Pantelis Z. Hadjipantelis <pantelis@ucdavis.edu>

References

Dabrowska, D. M. and Doksun K. A. (1988) Partial Likelihood in Transformation Models with Censored Data. *Scandinavian Journal of Statistics* **15**, 1–23.

Ding, J. and Wang, J. L. (2008) Modeling longitudinal data with nonparametric multiplicative random effects jointly with survival data. *Biometrics* **64**, 546–556.

Tsiatis, A. A. and Davidian, M. (2004) Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica* **14**, 809–834.

Xu, C., Baines, P. D. and Wang, J. L. (2014) Standard error estimation using the EM algorithm for the joint modeling of survival and longitudinal data. *Biostatistics* **15**, 731–744

Xu, C., Hadjipantelis, P. Z. and Wang, J. L. (2020) Semiparametric joint modeling of survival and longitudinal data: the R package JSM. *Journal of Statistical Software* <doi:10.18637/jss.v093.i02>.

Zeng, D. and Lin, D. (2007) Maximum likelihood estimation in semiparametric regression models with censored data. *Journal of the Royal Statistical Society: Series B* **69**, 507–564.

See Also

jmodelMultObject, lme, coxph, Surv, bs

Examples

```
# linear mixed-effects model fit where the fixed effect is modeled by
# quadratic B-splie basis with no internal knots
fitLME <- lme(log(serBilir) ~ bs(obstime, degree = 2, Boundary.knots = c(0, 15)),
              random = ~ 1 | ID, data = pbc)
# Cox proportional hazards model fit with a single time-independent covariate
fitCOX <- coxph(Surv(start, stop, event) ~ drug, data = pbc, x = TRUE)

# joint model fit which assumes the Cox proportional hazards model for the survival process
# and NMRE for the longitudinal process. Use 'max.iter = 25', 'nknot = 3' and
# the 'PFDS' method to calculate standard error estimates as a quick toy example
fitJTMult.ph <- jmodelMult(fitLME, fitCOX, pbc, timeVarY = "obstime",
                           control = list(SE.method = 'PFDS', max.iter = 25, nknot = 3))
summary(fitJTMult.ph)

## Not run:
# joint model fit with the default control
fitJTMult.ph2 <- jmodelMult(fitLME, fitCOX, pbc, timeVarY = "obstime")
summary(fitJTMult.ph2)
# joint model fit where the survival and longitudinal processes only share
# the same random effect
fitJTMult.ph3 <- jmodelMult(fitLME, fitCOX, pbc, model = 2, timeVarY = "obstime")
summary(fitJTMult.ph3)

# joint model fit which assumes the proportional odds model for the survival process
# and NMRE model for the longitudinal process
fitJTMult.po <- jmodelMult(fitLME, fitCOX, pbc, rho = 1, timeVarY = "obstime")
summary(fitJTMult.po)
# joint model fit where the survival and longitudinal processes only share
# the same random effect
fitJTMult.po2 <- jmodelMult(fitLME, fitCOX, pbc, model = 2, rho = 1, timeVarY = "obstime")
summary(fitJTMult.po2)

# allow baseline covariates in the NMRE model for the longitudinal process
```

```
fitLME2 <- lme(log(serBilir) ~ drug + bs(obstime, degree = 2, Boundary.knots = c(0, 15)),
               random = ~1 | ID, data = pbc)
fitJTMult.ph4 <- jmodelMult(fitLME2, fitCOX, pbc, timeVarY = "obstime")
summary(fitJTMult.ph4)

# Cox proportional hazards model fit with a time-dependent covariate
fitCOX2 <- coxph(Surv(start, stop, event) ~ drug + as.numeric(drug) : obstime,
                 data = pbc, x = TRUE)
# joint model fit in which \code{timeVarT} must be specified
fitJTMult.ph5 <- jmodelMult(fitLME, fitCOX2, pbc, timeVarY = "obstime", timeVarT = 'obstime',
                            control = list(max.iter = 300))
summary(fitJTMult.ph5)

## End(Not run)
```

`jmodelMultObject` *Fitted jmodelMult Object*

Description

An object returned by the `jmodelMult` function, inheriting from class `jmodelMult` and representing a fitted joint model for survival and longitudinal data. Objects of this class have methods for the generic functions `AIC`, `BIC`, `logLik`, `print`, `summary`, and `vcov`.

Value

The following components must be included in a legitimate `jmodelMult` object.

`coefficients` a list with the estimated parameters. The list is consist of the following components:

gamma the vector of estimated coefficients for the B-spline basis functions in the nonparametric multiplicative random effects model.

phi the vector of estimated coefficients for the covariates other than the covariate associated with the longitudinal process in the survival model.

alpha the estimated coefficient for the covariate associated with the longitudinal process in the survival model.

Ysigma the estimated measurement error standard deviation for the linear mixed-effects model.

Bsigma the estimated variance-covariance matrix of the random effects.

lamb a numeric matrix with two columns: the first column contains the unique observed survival times in ascending order; the second column contains the corresponding estimated baseline hazard values.

`Vcov` the variance-covariance matrix evaluated at the estimated parameter values.

`logLik` the log-likelihood (the joint likelihood) value.

`est.bi` the estimated values for the random effects

call	a list containing an image of the jmodelTM call that produced the object.
numIter	the number of iterations used in the EM algorithm.
convergence	the convergence indicator: if "failure", usually more iterations are required.
control	the value of the control argument passed to jmodelTM.
time.SE	the CPU time used to compute the standard error estimates, i.e. the time use to compute the variance-covariance matrix for the parameter estimates.
N	the total number of repeated measurements for the longitudinal outcome.
n	the number of sample units.
d	the censoring indicator: 0 denotes censored survival time; 1 denotes observed survival time.
rho	the transformation parameter used for the survival model.

Author(s)

Cong Xu <helenxu1112@gmail.com> Pantelis Z. Hadjipantelis <pantelis@ucdavis.edu>

See Also

jmodelMult

---

jmodelTM *Semiparametric Joint Models for Survival and Longitudinal Data*

Description

This function applies a maximum likelihood approach to fit the semiparametric joint models of survival and normal longitudinal data. The survival model is assumed to come from a class of transformation models, including the Cox proportional hazards model and the proportional odds model as special cases. The longitudinal process is modeled by liner mixed-effects models.

Usage

```
jmodelTM(fitLME, fitCOX, data, model = 1, rho = 0, timeVarY = NULL,
         timeVarT = NULL, control = list(), ...)
```

Arguments

fitLME	an object inheriting from class lme representing a fitted linear mixed-effects model. See **Note**.
fitCOX	an object inheriting from class coxph representing a fitted Cox proportional hazards regression model. Specifying x = TRUE is required in the call to coxph() to include the design matrix in the object fit. See **Note**.
data	a data.frame containing all the variables included in the joint modeling. See **Note**.

`model` an indicator specifying the dependency between the survival and longitudinal outcomes. Default is 1. See **Details**.

`rho` a nonnegative real number specifying the transformation model you would like to fit. Default is 0, i.e. the Cox proportional hazards model. See **Details**.

`timeVarY` a character string indicating the time variable in the linear mixed-effects model. See **Examples**.

`timeVarT` a character string indicating the time variable in the `coxph` object. Normally it is `NULL`. See **Note** and **Examples**.

`control` a list of control values for the estimation algorithm with components:

tol.P tolerance value for convergence in the parameters with default value 1e-03. See **Details**.

tol.L tolerance value for convergence in the log-likelihood with default value 1e-06. See **Details**.

max.iter the maximum number of EM iterations with default value 250.

SE.method a character string specifying the standard error estimation method. Default is `"PRES"`. See **Details** and **Note**.

delta a positive value used for numerical differentiation in the `SE.method`. Default is 1e-05 if `"PRES"` is used and 1e-03 otherwise. See **Details**.

nknot the number of Gauss-Hermite quadrature knots used to approximate the integrals over the random effects. Default is 9 and 7 for one- and two-dimensional integration, respectively, and 5 for those with higher dimensions.

`...` additional options to be passed to the `control` argument.

Details

The `jmodelTM` function fits joint models for survival and longitudinal data. Linear mixed-effects models are assumed for the longitudinal processes. With the Cox proportional hazards model and the proportional odds model as special cases, a general class of transformation models are assumed for the survival processes. The baseline hazard functions are left unspecified, i.e. no parametric forms are assumed, thus leading to semiparametric models. For detailed model formulation, please refer to Xu, Baines and Wang (2014).

The longitudinal model is written as

$$Y_i(t) = \mu_i(t) + \varepsilon_i(t) = \mathbf{X}_i^\top(t)\boldsymbol{\beta} + \mathbf{Z}_i^\top(t)\mathbf{b}_i + \varepsilon_i(t).$$

, then the linear predictor for the survival model is expressed as

$$\eta(t) = \mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha\mu_i(t),$$

indicating that the entire longitudinal process (free of error) enters the survival model as a covariate. If other values are assigned to the `model` argument, the linear predictor for the surival model is then expressed as

$$\eta(t) = \mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha\mathbf{Z}_i^\top(t)\mathbf{b}_i,$$

suggesting that the survival and longitudinal models only share the same random effects.

The survival model is written as

$$\Lambda(t|\eta(t)) = G\left[\int_0^t \exp\eta(s)d\Lambda_0(s)\right],$$

where $G(x) = \log(1+\rho x)/\rho$ with $\rho \geq 0$ is the class of logarithmic transfomrations. If `rho = 0`, then $G(x) = x$, yielding the Cox proportional hazards model. If `rho = 1`, then $G(x) = \log(1+x)$, yielding the proportional odds model. Users could assign any nonnegative real value to `rho`.

An expectation-maximization (EM) algorithm is implemented to obtain parameter estimates. The convergence criterion is either of (i) $\max\{|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}|/(|\boldsymbol{\theta}^{(t-1)}| + .Machine\$double.eps \times 2)\} < tol.P$, or (ii) $|L(\boldsymbol{\theta}^{(t)}) - L(\boldsymbol{\theta}^{(t-1)})|/(|L(\theta^{(t-1)})| + .Machine\$double.eps \times 2) < tol.L$, is satisfied. Here $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t-1)}$ are the vector of parameter estimates at the t-th and $(t-1)$-th EM iterations, respectively; $L(\boldsymbol{\theta})$ is the value of the log-likelihood function evaluated at $\boldsymbol{\theta}$. Users could specify the tolerance values `tol.P` and `tol.L` through the `control` argument.

For standard error estimation for the parameter estimates, three methods are provided, namely `"PRES"`, `"PFDS"` and `"PLFD"` (detailed information are referred to Xu, Baines and Wang (2014)). In the `control` argument, if `SE.method = "PRES"`, numerically differentiating the profile Fisher score vector with Richardson extrapolation is applied; if `SE.method = "PFDS"`, numerically differentiating the profile Fisher score vector with forward difference is applied; if `SE.method = "PLFD"`, numerially (second) differentiating the profile likelihood with forward difference is applied. Generally, numerically differentiating a function $f(x)$ (an arbitrary function) with forward difference is expressed as

$$f'(x) = \frac{f(x+\delta) - f(x)}{\delta},$$

and that with Richardson extrapolation is expressed as

$$f'(x) = \frac{f(x-2\delta) - 8f(x-\delta) + 8f(x+\delta) - f(x+2\delta)}{12\delta}.$$

Users could specify the value of δ through the `delta` item in the `control` argument.

Value

See `jmodelTMObject` for the components of the fit.

Note

1. Currently, `jmodelTM()` could only handle the `fitLME` object with a simple random-effects structure (only the `pdDiag()` class). Moreover, the within-group correlation and heteroscedasticity structures in the `fitLME` object (i.e. the `correlation` and `weights` argument of `lme()`) are ignored.

2. The `data` argument in `jmodelTM()`, `lme()` and `coxph()` should be the same data frame.

3. For the `fitCOX` object, only the $W_i(t)$ in the linear predictor $\eta(t)$ for the survial model (see **Details**) should be involved in the `formula` argument of `coxph{}`. Since `coxph()` uses the same data frame as `lme()` does, a time-dependent Cox model must be fitted by `coxph()` although $W_i(t)$ may only contain time-independent covariates. See **Examples**.

4. If $W_i(t)$ in the linear predictor $\eta(t)$ for the survial model (see **Details**) does involve time-dependent covariate, then `timeVarT` must specify the name of the time variable involved. See **Examples**.

5. The standard error estimates are obtained by numerical approximations which is naturally subject to numerical errors. Therefore, in extreme cases, there may be NA values for part of the standard error estimates.

Author(s)

Cong Xu <helenxu1112@gmail.com> Pantelis Z. Hadjipantelis <pantelis@ucdavis.edu>

References

Dabrowska, D. M. and Doksun K. A. (1988) Partial Likelihood in Transformation Models with Censored Data. *Scandinavian Journal of Statistics* **15**, 1–23.

Tsiatis, A. A. and Davidian, M. (2004) Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica* **14**, 809–834.

Wulfsohn, M. S. and Tsiatis, A. A. (1997) A joint model for survival and longitudinal data measured with error. *Biometrics* **53**, 330–339.

Xu, C., Baines, P. D. and Wang, J. L. (2014) Standard error estimation using the EM algorithm for the joint modeling of survival and longitudinal data. *Biostatistics* **15**, 731–744.

Xu, C., Hadjipantelis, P. Z. and Wang, J. L. (2020) Semiparametric joint modeling of survival and longitudinal data: the R package JSM. *Journal of Statistical Software* <doi:10.18637/jss.v093.i02>.

Zeng, D. and Lin, D. (2007) Maximum likelihood estimation in semiparametric regression models with censored data. *Journal of the Royal Statistical Society: Series B* **69**, 507–564.

See Also

jmodelTMObject, lme, coxph, Surv

Examples

```
# linear mixed-effects model fit with random intercept
fitLME <- lme(sqrt(CD4) ~ obstime + I(obstime ^ 2) + drug : obstime + drug : I(obstime ^ 2),
              random = ~ 1 | ID, data = aids)
# Cox proportional hazards model fit with a single time-independent covariate
fitCOX <- coxph(Surv(start, stop, event) ~ drug, data = aids, x = TRUE)

# joint model fit which assumes the Cox proportional hazards model for the survival process
# Use 'max.iter = 5', 'nknot = 3' and the 'PFDS' method to calculate standard
# error estimates as a quick toy example
fitJT.ph <- jmodelTM(fitLME, fitCOX, aids, timeVarY = 'obstime',
                     control = list(SE.method = 'PFDS', max.iter = 5, nknot = 3))
summary(fitJT.ph)

## Not run:
# joint model fit with the default control
fitJT.ph2 <- jmodelTM(fitLME, fitCOX, aids, timeVarY = 'obstime')
summary(fitJT.ph2)
# joint model fit where the survival and longitudinal processes only share
# the same random effect
fitJT.ph3 <- jmodelTM(fitLME, fitCOX, aids, model = 2, timeVarY = 'obstime')
```

```
summary(fitJT.ph3)

# joint model fit which assumes the proportional odds model for the survival process
fitJT.po <- jmodelTM(fitLME, fitCOX, aids, rho = 1, timeVarY = 'obstime')
summary(fitJT.po)
# joint model fit where the survival and longitudinal processes only share
# the same random effect
fitJT.po2 <- jmodelTM(fitLME, fitCOX, aids, model = 2, rho = 1, timeVarY = 'obstime')
summary(fitJT.po2)

# linear mixed-effects model fit with random intercept and random slope
fitLME2 <- lme(sqrt(CD4) ~ drug + obstime + I(obstime ^ 2) + drug : obstime +
               drug : I(obstime ^2), random = ~ obstime | ID, data = aids)
# Cox proportional hazards model fit with a time-dependent covariate
fitCOX2 <- coxph(Surv(start, stop, event) ~ drug + as.numeric(drug) : obstime,
                 data = aids, x = TRUE)
# joint model fit in which \code{timeVarT} must be specified
fitJT.ph4 <- jmodelTM(fitLME2, fitCOX2, aids, timeVarY = 'obstime', timeVarT = 'obstime')
summary(fitJT.ph4)

## End(Not run)
```

`jmodelTMObject` *Fitted jmodelTM Object*

Description

An object returned by the `jmodelTM` function, inheriting from class `jmodelTM` and representing a fitted joint model for survival and longitudinal data. Objects of this class have methods for the generic functions `AIC`, `BIC`, `logLik`, `print`, `summary`, and `vcov`.

Value

The following components must be included in a legitimate `jmodelTM` object.

`coefficients` a list with the estimated parameters. The list is consist of the following components:

beta the vector of estimated coefficients for the fixed effects in the linear mixed-effects model.

phi the vector of estimated coefficients for the covariates other than the covariate associated with the longitudinal process in the survival model.

alpha the estimated coefficient for the covariate associated with the longitudinal process in the survival model.

Ysigma the estimated measurement error standard deviation for the linear mixed-effects model.

BSigma the estimated variance-covariance matrix of the random effects.

lamb a numeric matrix with two columns: the first column contains the unique observed survival times in ascending order; the second column contains the corresponding estimated baseline hazard values.

Vcov	the variance-covariance matrix evaluated at the estimated parameter values.
logLik	the log-likelihood (the joint likelihood) value.
est.bi	the estimated values for the random effects
call	a list containing an image of the jmodelTM call that produced the object.
numIter	the number of iterations used in the EM algorithm.
convergence	the convergence indicator: if "failure", usually more iterations are required.
control	the value of the control argument passed to jmodelTM.
time.SE	the CPU time used to compute the standard error estimates, i.e. the time use to compute the variance-covariance matrix for the parameter estimates.
N	the total number of repeated measurements for the longitudinal outcome.
n	the number of sample units.
d	the censoring indicator: 0 denotes censored survival time; 1 denotes observed survival time.
rho	the transformation parameter used for the survival model.

Author(s)

Cong Xu <helenxu1112@gmail.com> Pantelis Z. Hadjipantelis <pantelis@ucdavis.edu>

See Also

jmodelTM

---

liver *Prednisone versus Placebo in Liver Cirrhosis Patients*

Description

A randomized control trial in which both survival and longitudinal data were collected to examine the development of prothrombin index over time and its relationship with the survival outcome. 488 patients were randomly allocated to prednisone (251) or placebo (237) and followed until death or end of the study.

Format

A data frame with 2968 observations on the following 9 variables.

ID patient ID, there are 488 patients in total.

Time survival time, i.e. time to death or censoring.

death death indicator: 0 denotes censoring; 1 denotes death.

obstime time points at which the longitudinal measurements, i.e. prothrombin index, are recorded.

proth prothrombin index measured at obstime.

Trt treatment indicator with two levels: placebo and prednisone.

`start` same with `obstime`, starting time of the interval which contains the time of the prothrombin index measurement.

`stop` ending time of the interval which contains the time of the prothrombin index measurement.

`event` event indicator suggesting whether the event-of-interest, i.e. death, happens in the interval given by `start` and `stop`.

Source

Andersen, P. K., Borgan O., Gill, R. D. and Kieding, N. (1993) *Statistical Models Based on Counting Processes*. New York: Springer.

References

Henderson, R., Diggle, P. and Dobson, A. (2002) Identification and efficacy of longitudinal markers for survival. *Biostatistics* **3**, 33–50

See Also

`liver.long`, `liver.surv`

Examples

```
head(liver)
```

`liver.long` *Prednisone versus Placebo in Liver Cirrhosis Patients - Longitudinal Data*

Description

A randomized control trial in which both survival and longitudinal data were collected to examine the development of prothrombin index over time and its relationship with the survival outcome. 488 patients were randomly allocated to prednisone (251) or placebo (237) and followed until death or end of the study. `liver.long` only contains the longitudinal data of the trial, with one row per prothrombin index measurement.

Format

A data frame with 2968 observations on the following 3 variables.

`ID` patient ID, there are 488 patients in total.

`obstime` time points at which the longitudinal measurements, i.e. prothrombin index, are recorded.

`proth` prothrombin index measured at `obstime`.

Source

Andersen, P. K., Borgan O., Gill, R. D. and Kieding, N. (1993) *Statistical Models Based on Counting Processes*. New York: Springer.

References

Henderson, R., Diggle, P. and Dobson, A. (2002) Identification and efficacy of longitudinal markers for survival. *Biostatistics* **3**, 33–50

See Also

liver, liver.surv

Examples

```
head(liver.long)
```

---

liver.surv *Prednisone versus Placebo in Liver Cirrhosis Patients - Survival Data*

Description

A randomized control trial in which both survival and longitudinal data were collected to examine the development of prothrombin index over time and its relationship with the survival outcome. 488 patients were randomly allocated to prednisone (251) or placebo (237) and followed until death or end of the study. liver.surv only contains the survival data of the trial, with one row per patient.

Format

A data frame with 488 observations on the following 4 variables.

ID patient ID, there are 488 patients in total.

Time survival time, i.e. time to death or censoring.

death death indicator: 0 denotes censoring; 1 denotes death.

Trt treatment indicator with two levels: placebo and prednisone.

Source

Andersen, P. K., Borgan O., Gill, R. D. and Kieding, N. (1993) *Statistical Models Based on Counting Processes*. New York: Springer.

References

Henderson, R., Diggle, P. and Dobson, A. (2002) Identification and efficacy of longitudinal markers for survival. *Biostatistics* **3**, 33–50

See Also

liver, liver.long

Examples

```
head(liver.surv)
```

pbc *Mayo Clinic Primary Biliary Cirrhosis Data*

Description

A randomized control trial from Mayo Clinic in which both survival and longitudinal data were collected from 1974 to 1984 to study the progression of primary biliary cirrhosis.

Format

A data frame with 1945 observations on the following 16 variables.

`ID` patient ID, there are 312 patients in total.

`Time` survival time (in years), i.e. time to death, transplantion or censoring.

`death` death indicator: 0 denotes transplantion or censoring; 1 denotes death.

`obstime` time points at which the longitudinal measurements, e.g. serum bilirubin, albumin and alkaline phosphatase, are recorded.

`serBilir` serum bilirubin measured at `obstime` (mg/dl).

`albumin` albumin measured at `obstime` (gm/dl).

`alkaline` alkaline phosphatase measured at `obstime` (U/litter).

`platelets` platelets per cubic measured at `obstime` (ml/1000).

`drug` drug indicator with two levels: `placebo` and `D-penicil`.

`age` age of patient at study entry.

`gender` gender indicator with two levels: `male` and `female`.

`ascites` ascites indicator with two levels: `No` and `Yes`.

`hepatom` hepatomegaly indicator with two levels: `No` and `Yes`.

`start` same with `obstime`, starting time of the interval which contains the time of the logitudinal measurements.

`stop` ending time of the interval which contains the time of the longitudinal measurements.

`event` event indicator suggesting whether the event-of-interest, i.e. death, happens in the interval given by `start` and `stop`.

Source

https://lib.stat.cmu.edu/datasets/pbcseq

Fleming, T. and Harrington, D. (1991) *Counting Processes and Survival Analysis*. Wiley, New York.

References

Murtaugh, P. A., Dickson, E. R., Van Dam, G. M., Malincho, M., Grambsch, P. M., Langworthy, A. L., and Gips, C. H. (1994) Primary biliary cirrhosis: Prediction of short-term survival based on repeated patient visits. *Hepatology* **20**, 126–134.

Therneau, T. and Grambsch, P. (2000) *Modeling Survival Data: Extending the Cox Model*. New York: Springer.

Ding, J. and Wang, J. L. (2008) Modeling longitudinal data with nonparametric multiplicative random effects jointly with survival data. *Biometrics* **64**, 546–556.

Examples

```
head(pbc)
```

`ranef` *Extract Random Effects for Joint Models*

Description

`ranef` is a generic function which extracts random effects from objects returned by `jmodelTM()` or `jmodelMult()`.

Usage

```
## S3 method for class 'jmodelTM'
ranef(object, ...)
## S3 method for class 'jmodelMult'
ranef(object, ...)
```

Arguments

`object`	an object inheriting from class `jmodelTM` or `jmodelMult`.
`...`	additional arguments required. None is used in this method.

Value

A numeric matrix with rows denoting the subjects and columns the random effects.

Author(s)

Cong Xu <helenxu1112@gmail.com>

Examples

```
## Not run:
fitLME <- lme(proth ~ Trt * obstime, random = ~ 1 | ID, data = liver)
fitCOX <- coxph(Surv(start, stop, event) ~ Trt, data = liver, x = TRUE)
fitJT.ph <- jmodelTM(fitLME, fitCOX, liver, timeVarY = 'obstime')

# random effect for the joint model
ranef(fitJT.ph)

## End(Not run)
```

`residuals` *Extract Residuals for Joint Models*

Description

`residuals` is a generic function which extracts residuals from objects returned by `jmodelTM()` or `jmodelMult()`.

Usage

```
## S3 method for class 'jmodelTM'
residuals(object, process = c("Longitudinal", "Survival"),
            type = c("Marginal", "Conditional", "Standardized-Marginal",
            "Standardized-Conditional"), ...)
## S3 method for class 'jmodelMult'
residuals(object, process = c("Longitudinal", "Survival"),
            type = c("Marginal", "Conditional", "Standardized-Marginal",
            "Standardized-Conditional"), ...)
```

Arguments

`object`	an object inheriting from class `jmodelTM` or `jmodelMult`.
`process`	for which process the residuals are calculated, i.e. the longitudinal or the survival process.
`type`	what type of residuals to calculate for each process. See **Details**.
`...`	additional arguments required. None is used in this method.

Details

We have implemented the residual calculation for `process = "Longitudinal"` but not for `process = "Survival"` yet as they are not well defined under the joint modeling setting. There are four types of residuals depending on whether to compute the values conditional on the random effects and whether to standardize the residuals. Please refer to Nobre and Single (2007) for details.

With `type = "Marginal"`, the residuals are $\varepsilon_{ij} = Y_{ij} - \mathbf{X}_{ij}^{\top}\boldsymbol{\beta}$ for objects returned by `jmodelTM()` and $\varepsilon_{ij} = Y_{ij} - \mathbf{B}^{\top}(t_{ij})\boldsymbol{\gamma}$ for objects returned by `jmodelMult()`. With `type = "Conditional"`,

the residuals are $\varepsilon_{ij} = Y_{ij} - \mathbf{X}_{ij}^{\top}\boldsymbol{\beta} - \mathbf{Z}_{ij}^{\top}\mathbf{b}_i$ for objects returned by `jmodelTM()` and $\varepsilon_{ij} = Y_{ij} - b_i \times \mathbf{B}^{\top}(t_{ij})\boldsymbol{\gamma}$ for objects returned by `jmodelMult()`. If `type = "Standardized-Marginal"` or `type = "Standardized-Conditional"`, the above defined residuals are divided by the estimated standard deviation of the corresponding error term.

Value

A numerc vector of residual values.

Author(s)

Cong Xu <helenxu1112@gmail.com>

References

Nobre, J. S. and Singer, J. M. (2007) Residuals analysis for linear mixed models. *Biometrical Jounral* **49**(6), 863–875.

See Also

`fitted.jmodelTM`, `fitted.jmodelMult`

Examples

```
## Not run:
fitLME <- lme(proth ~ Trt * obstime, random = ~ 1 | ID, data = liver)
fitCOX <- coxph(Surv(start, stop, event) ~ Trt, data = liver, x = TRUE)
fitJT.ph <- jmodelTM(fitLME, fitCOX, liver, timeVarY = 'obstime')

# residuals for the longitudinal process
residuals(fitJT.ph, type = "Standardized-Conditional")

## End(Not run)
```

Index