

Package ‘JointModel’

October 12, 2022

Title Semiparametric Joint Models for Longitudinal and Counting Processes

Version 1.0

Date 2016-06-01

Author Sehee Kim <seheek@umich.edu>

Maintainer Sehee Kim <seheek@umich.edu>

Description Joint fit of a semiparametric regression model for longitudinal responses and a semiparametric transformation model for time-to-event data.

Depends R (>= 3.2.0), lme4, survival, splines, statmod

License GPL-3

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2016-06-09 05:49:19

R topics documented:

dropout	2
jplm	3
pred.jplm.cumhaz	5
pred.jplm.nonlinear	7
prostate	8
Index	9

dropout	<i>Prostate Cancer Data: Part 2 - a simulated example of study drop-out process</i>
---------	---

Description

A simulated data set used in Kim et al. (2016) to illustrate the JPLM method. This data set was generated under settings mimicking the prostate cancer study. For detailed data generation settings, see Kim et al. (2016).

Format

A data frame with 100 observations (n = 100 patients) on the following 4 variables.

ID2 a numeric vector of patient ID.

Status a numeric (binary) vector indicating whether the study drop-out time (end of follow-up) was informative.

DropTime a numeric vector of the end of follow-up.

logPSA.base2 a numeric vector of $\log(\text{baseline PSA} + 0.1)$.

Source

The prostate cancer data have been previously studied by Proust-Lima et al. (2008) and Taylor et al. (2013), among others.

References

Kim, S., Zeng, D., Taylor, J. M. G. (2016) Joint partially linear model for longitudinal data with informative drop-outs. *Under revision* **0**, 000-000.

Proust-Lima, C., Taylor, J. M. G., Williams, S. G., Ankerst, D. P., Liu, N., Kestin, L. L., Bae, K., and Sandler, H. M. (2008) Determinants of change in prostate-specific antigen over time and its association with recurrence after external beam radiation therapy for prostate cancer in five large cohorts. *International Journal of Radiation Oncology Biology Physics* **72**, 782-791.

Taylor, J. M. G., Part, Y., Ankerst, D. P., Proust-Lima, C., Williams, S., Kestin, L., Bae, K., Pickles, T., and Sandler, H. (2013) Real-time individual predictions of prostate cancer recurrence using joint models. *Biometrics* **69**, 206-213.

jplm

*Joint Partially Linear Model for Longitudinal and Time-to-Event Data***Description**

This function fits jointly a partially linear model for normal longitudinal responses and a semiparametric transformation model for time-to-event data using a sieve maximum likelihood approach (Kim et al., 2016).

Usage

```
jplm(formula.lm.y, nlm.par=NULL, data.y,
      formula.surv.fixed, formula.frailty,
      id.vec=NULL, transf.par=0, data.surv,
      degree=3, n.knots=3, CovEst=TRUE, maxiter=200,
      epsilon=5e-04,...)
```

Arguments

<code>formula.lm.y</code>	a formula inheriting from class <code>lmer</code> . The right side of the formula describes both the linear fixed effects and random effects part of the longitudinal model. Random-effects terms are distinguished by one vertical bar (<code> </code>) separating expressions for design matrices from grouping factors. Only one vertical var is allowed.
<code>nlm.par</code>	a vector of nonlinear effect covariate.
<code>data.y</code>	a <code>data.frame</code> containing the variables named in <code>formula.lm.y</code> .
<code>formula.surv.fixed</code>	a formula inheriting from class <code>coxph</code> . That is, the response must be a survival object as returned by the <code>Surv(,)</code> function, and the right side of formula must include only fixed effects covariates.
<code>formula.frailty</code>	a formula with the right side describing frailty term of the survival model. The left side of the formula must leave in blank, and the intercept term will be included unless specified by <code>~-1+...</code> or <code>~0+...</code>
<code>id.vec</code>	a vector containing subject ID corresponding to <code>formula.frailty</code> .
<code>transf.par</code>	a non-negative value of transformation parameter applied to the cumulative hazard function. <code>transf.par=0</code> will fit a proportional hazards model, while <code>transf.par=1</code> will fit a proportional odds model. Default is 0.
<code>data.surv</code>	a <code>data.frame</code> containing the variables named in the <code>formula.surv.fixed</code> .
<code>degree</code>	degree of the sieve polynomial. Default is 3 for cubic splines.
<code>n.knots</code>	the number of interior knot points for the B-spline approximation of the nonlinear effect. Default is 3.
<code>CovEst</code>	logical value; if <code>TRUE</code> , the covariate matrix of all the model parameters are estimated.

maxiter	the maximum number of EM iterations. Default is 200.
epsilon	tolerance value in the Newton-Raphson algorithm used to update the parameters in the M-step. Default is 5e-04.
...	other arguments

Details

Function `jplm` fits joint models for longitudinal and survival data. Viewing time-to-drop-out as an event process, the same joint models also can be fitted to longitudinal data with informative drop-outs. For more detailed formulation of these models, refer to Kim et al. (2016).

For the longitudinal model specification, all linear effects part (including fixed and random) should be described in `formula.lm.y`, while a non-linear fixed effect component should be specified in the `nlm.par` argument.

For the survival model specification, the fixed effects component should be described by the argument `formula.surv.fixed`, whereas the random effects (frailty) component should be described in `formula.frailty`. The current version assumes all random effects coefficients are the same. For example, the argument `formula.frailty=~ 1+time` implies a regression model with $\phi(b_1 + b_2 * time)$, where b_1 and b_2 represent the random intercept and slope terms and ϕ is one dimensional coefficient corresponding to both b_1 and b_2 .

A logarithmic transformation can be applied to the cumulative hazard function by varying the value of `transf.par`. Specifically, the transformation function takes the form of: $H(x) = 0$ if `transf.par=0`; otherwise $H(x) = \log(1 + K)/K$ with `transf.par=K`. `transf.par=0` will fit a proportional hazards model, while `transf.par=1` will fit a proportional odds model.

For AIC- or BIC-based model selection, `CovEst=FALSE` is strongly recommended.

Value

<code>coef.lm.y</code>	the vector of linear coefficients in the longitudinal model.
<code>coef.nlm.y</code>	the vector of sieve coefficients corresponding to B-spline approximation of the nonlinear effect in the longitudinal model.
<code>var.resid</code>	the variance estimate of longitudinal response residuals.
<code>raneff.vcomp</code>	the vector of estimates of random effects variance component, corresponding to random intercept, random slope, and their correlation.
<code>coef.fixed.surv</code>	the vector of fixed coefficients in the survival model.
<code>coef.frailty.surv</code>	value of random (frailty) coefficient in the survival model.
<code>lambdas</code>	the vector of jump sizes of the baseline cumulative hazard function, corresponding to the ordered observed event times.
<code>loglik</code>	value of the log-likelihood with the final values of the coefficients.
AIC	value of AIC with the final values of the coefficients.
BIC	value of BIC with the final values of the coefficients.
<code>degree</code>	degree used for the sieve polynomial.

n.knots	number of interior knot points used for the B-spline approximation of the non-linear effect.
K	the transformation parameter.
covy	the variance matrix of all the parameters in the longitudinal model, corresponding to the estimates of coef.lm.y,coef.nlm.y, and var.resid.
covb	the variance matrix of all the parameters in raneff.vcomp.
covt	the variance matrix of all the parameters in the survival model, corresponding to the estimates of coef.fixed.surv,coef.frailty.surv, and lambdas.

Author(s)

Sehee Kim

References

Kim, S., Zeng, D., Taylor, J. M. G. (2016) Joint partially linear model for longitudinal data with informative drop-outs. *Under revision* **0**, 000-000.

See Also

[pred.jplm.nonlinear](#), [pred.jplm.cumhaz](#)

Examples

```
# a simulated data set of longitudinal responses
attach(prostate)
# a simulated data set of time-to-event (e.g., drop-out process)
attach(dropout)

# joint fit of a partially linear model and a proportional hazards model
# with a subject-specific random intercept and random slope
fit0 <- jplm(logPSA.postRT ~ logPSA.base + (1 + VisitTime|ID),
            nlm.par=prostate$VisitTime, data.y=prostate,
            Surv(DropTime, Status) ~ logPSA.base2,
            formula.frailty= ~ 1 + DropTime,
            id.vec=dropout$ID2, transf.par=0, data.surv=dropout)

summary(fit0)
```

pred.jplm.cumhaz	<i>Predict the baseline cumulative hazard function at any given time point</i>
------------------	--

Description

This function calculates a predicted baseline cumulative hazard function, evaluated at given time points.

Usage

```
pred.jplm.cumhaz(object, at=NULL, CI=FALSE)
```

Arguments

object	a Joint Model fit object, i.e., the result of <code>jplm</code> .
at	a vector of fixed time points to be evaluated.
CI	logical value; if TRUE, a 95% pointwise confidence interval is returned.

Value

If `CI=FALSE`, it returns a numeric vector of predicted cumulative hazard values at `at=` . If `CI=TRUE`, it returns a numeric vector of predicted cumulative hazard values, the standard error estimate of the predicted value, and its lower and upper 95% pointwise confidence interval.

Author(s)

Sehee Kim

References

Kim, S., Zeng, D., Taylor, J.M.G. (2016) Joint partially linear model for longitudinal data with informative drop-outs. *Under revision* **0**, 000-000.

See Also

[jplm](#)

Examples

```
# a simulated data set of longitudinal responses
attach(prostate)
# a simulated data set of drop-out process (or, time-to-event)
attach(dropout)

# joint fit of a partially linear model and a proportional odds model
# with a subject-specific random intercept and random slope
fit1 <- jplm(logPSA.postRT ~ logPSA.base + (1 + VisitTime|ID),
            nlm.par=prostate$VisitTime, data.y=prostate,
            Surv(DropTime, Status) ~ logPSA.base2,
            formula.frailty= ~ 1 + DropTime,
            id.vec=dropout$ID2, transf.par=1, data.surv=dropout)

# Evaluate at 20,...,80 percent of the maximum observed survival time
pts <- c(0.2, 0.4, 0.6, 0.8)*max(dropout$DropTime)
pred.jplm.cumhaz(fit1, at=pts)
out <- pred.jplm.cumhaz(fit1, at=pts, CI=TRUE)
out$Value
```

pred.jplm.nonlinear *Predict a smoothed nonlinear effect on the longitudinal response*

Description

This function calculates a predicted nonlinear effect function evaluated at given points.

Usage

```
pred.jplm.nonlinear(object, nlm.par, at=NULL, CI=FALSE)
```

Arguments

object	a Joint Model fit object, i.e., the result of <code>jplm</code> .
nlm.par	a vector of nonlinear effect covariate, as specified in <code>nlm.par=</code> of <code>jplm</code> .
at	a vector of fixed points to be evaluated.
CI	logical value; if TRUE, a 95% pointwise confidence interval is returned.

Value

If `CI=FALSE`, it returns a numeric vector of predicted nonlinear effect at `at=`, the standard error estimate of the predicted value, and test result based on the asymptotic normality. If `CI=TRUE`, it returns a numeric vector of predicted nonlinear effect, the standard error estimate of the predicted value, and its lower and upper 95% pointwise confidence interval.

Author(s)

Sehee Kim

References

Kim, S., Zeng, D., Taylor, J.M.G. (2016) Joint partially linear model for longitudinal data with informative drop-outs. *Under revision* **0**, 000-000.

See Also

[jplm](#)

Examples

```
# a simulated data set of longitudinal responses
attach(prostate)
# a simulated data set of time-to-event (e.g., drop-out process)
attach(dropout)

# joint fit of a partially linear model and a proportional odds model
# with a subject-specific random intercept and random slope
```

```

fit1 <- jplm(logPSA.postRT ~ logPSA.base + (1 + VisitTime|ID),
            nlm.par=prostate$VisitTime, data.y=prostate,
            Surv(DropTime, Status) ~ logPSA.base2,
            formula.frailty= ~ 1 + DropTime,
            id.vec=dropout$ID2, transf.par=1, data.surv=dropout)

# Evaluate at 20,...,80 percent of the maximum measurement time
pts <- c(0.2, 0.4, 0.6, 0.8)*max(prostate$VisitTime)
pred.jplm.nonlinear(fit1, prostate$VisitTime, at=pts)
out <- pred.jplm.nonlinear(fit1, prostate$VisitTime, at=pts, CI=TRUE)
out$Value

```

prostate

Prostate Cancer Data: Part 1 - a simulated example of longitudinal responses

Description

A simulated data set used in Kim et al. (2016) to illustrate the JPLM method. This data set was generated under settings mimicking the prostate cancer study. For detailed data generation settings, see Kim et al. (2016).

Format

A data frame with 697 observations ($n = 100$ patients) on the following 4 variables.

ID a numeric vector of patient ID.

logPSA.postRT a numeric vector containing Prostate-specific Antigen (PSA) levels after radiation therapy, i.e., $\log(\text{PSA}(t) + 0.1)$ observed at time t .

VisitTime a numeric vector of visiting time.

logPSA.base a numeric vector of $\log(\text{baseline PSA} + 0.1)$.

Source

The prostate cancer data have been previously studied by Proust-Lima et al. (2008) and Taylor et al. (2013), among others.

References

- Kim, S., Zeng, D., Taylor, J. M. G. (2016) Joint partially linear model for longitudinal data with informative drop-outs. *Under revision* **0**, 000-000.
- Proust-Lima, C., Taylor, J. M. G., Williams, S. G., Ankerst, D. P., Liu, N., Kestin, L. L., Bae, K., and Sandler, H. M. (2008) Determinants of change in prostate-specific antigen over time and its association with recurrence after external beam radiation therapy for prostate cancer in five large cohorts. *International Journal of Radiation Oncology Biology Physics* **72**, 782-791.
- Taylor, J. M. G., Part, Y., Ankerst, D. P., Proust-Lima, C., Williams, S., Kestin, L., Bae, K., Pickles, T., and Sandler, H. (2013) Real-time individual predictions of prostate cancer recurrence using joint models. *Biometrics* **69**, 206-213.

Index

* datasets

dropout, [2](#)

prostate, [8](#)

dropout, [2](#)

jplm, [3](#), [6](#), [7](#)

pred.jplm.cumhaz, [5](#), [5](#)

pred.jplm.nonlinear, [5](#), [7](#)

prostate, [8](#)