Package 'spef'

May 24, 2020

Title Semiparametric Estimating Functions

Version 1.0.9

Description Functions for fitting semiparametric regression models for panel count survival data. An overview of the package can be found in Wang and Yan (2011) <doi:10.1016/j.cmpb.2010.10.005> and Chiou et al. (2018) <doi:10.1111/insr.12271>.

Depends R (>= 3.4.0)

License GPL (>= 3)

URL http://github.com/stc04003/spef

BugReports http://github.com/stc04003/spef/issues

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports BB, SQUAREM, ggplot2, methods, sm, survival, plyr, nleqslv

NeedsCompilation yes

Author Sy Han (Steven) Chiou [aut, cre],
Xiaojing Wang [aut],
Jun Yan [aut]

Maintainer Sy Han (Steven) Chiou <schiou@utdallas.edu>

Repository CRAN

Date/Publication 2020-05-24 05:00:02 UTC

R topics documented:

spef-package *spef: Semiparametric Estimating Functions*

Description

`spef` is an R package that consists of a collection of functions for fitting semiparametric regression models for panel count survival data. Estimating procedures include: Wang-Yan's augmented estimating equations (AEE, AEEX), Huang-Wang-Zhang's method (HWZ), Zhang's maximum pseudolikelihood (MPL), Maximum pseudolikelihood with I-Splines (MPLs), Maximum likelihood with I-Splines (MLs), Sun-Wei's method (‘EE.SWa‘, ‘EE.SWb‘, ‘EE.SWc‘), Hu-Sun-Wei's method (‘EE.HSWc‘, ‘EE.HSWm‘), and accelerated mean model (‘AMM‘).

Author(s)

Maintainer: Sy Han (Steven) Chiou `<schiou@utdallas.edu>`

Authors:

- Xiaojing Wang
- Jun Yan

References

Chiou, S., Xu, G., Yan, J., and Huang, C.-Y. (2017). Semiparametric estimation of the accelerated mean model with panel count data under informative examination times. *Biometrics*, to appear. <doi: 10.1111/biom.12840>.

Huang, C.-Y., Wang, M., and Zhang, Y. (2006). Analysing panel count data with informative observation times. *Biometrika*, **93**(4), 763–776.

Hu, X. J., Sun, J. and Wei, L. J. (2003). Regression parameter estimation from panel counts. *Scandinavian Journal of Statistics*, **30**, 25–43.

Lu, M., Zhang, Y., and Huang, J. (2007). Estimation of the mean function with panel count data using monotone polynomial splines. *Biometrika*, **94**(3), 705–718.

Sun, J. and Wei, L. J. (2000). Regression analysis of panel count data with covariates-dependent observation and censoring times. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, **62**(2), 293–302.

Wang, X. and Yan, J. (2011). Fitting semiparametric regressions for panel count survival data with an R package spef. *Computer methods and programs in biomedicine* **104**(2), 278–285.

Wang, X. and Yan, J. (2013). Augmented estimating equations for semiparametric panel count regression with informative observation times and censoring time. *Statistica Sinica*, **23**(1), 359–381.

Zhang, Y. (2002). A Semiparametric pseudolikelihood estimation method for panel count data. *Biometrika*, **89**(1), 39–48.

See Also

Useful links:

- `http://github.com/stc04003/spef`
- Report bugs at `http://github.com/stc04003/spef/issues`

`bladTumor` *Bladder Tumors Cancer Recurrences*

Description

A data frame contains data on recurrences of bladder cancer, used by many people to demonstrate methodology for recurrent event modeling. The data was obtained by courtesy of Ying Zhang, containing records of 118 patients from three treatment arms: 48 are from the placebo arm, 37 are from the thiotepa arm, and the rest 33 are from the pyridoxine arm.

Usage

```
data(bladTumor)
```

Format

A `data.frame` contains the following columns:

`subject` patient id

`time` observation time

`count` cumulative number of tumors

`count2` number of new tumors since last observation time

`number` initial number of tumors (8=8 or more)

`size` size (cm) of largest initial tumor

`pyridoxine` dummy variable for pyridoxine arm

`thiotepa` dummy variable for thiotepa arm

Note

To our surprise, the two-treatment (placebo and thiotepa) subset of the full version `bladTumor` do not match the two-treatment version `blaTum`.

References

Byar, D. P. (1980). The Veterans administration study of chemoprophylaxis for recurrent stage I bladder tumors: Comparisons of placebo, pyridoxine, and topical thiotepa. *Bladder Tumors and Other Topics in Urological Oncology*, pp. 363–370. New York: Plenum.

Wellner, J. A. and Zhang, Y. (2007) Two likelihood-based semiparametric estimation methods for panel count data with covariates. *Annals of Statistics*, **35**(5), 2106–2142.

Lu, M., Zhang, Y. and Huang, J. (2009) Semiparametric estimation methods for panel count data using monotone B-Splines. *Journal of the American Statistical Association* **104**(487), 1060–1070.

See Also

blaTum

Examples

```
data(bladTumor)
## Plot bladder tumor data
p <- plot(with(bladTumor, PanelSurv(subject, time, count2)))
print(p)
```

blaTum *Bladder Tumors Cancer Recurrences*

Description

A data frame contains data on recurrences of bladder cancer, used by many people to demonstrate methodology for recurrent event modelling. This data set organized differently from bladTumor. The data contains records of 85 patients from two treatment arms: 48 are from the placebo arm, and the rest 37 are from the thiotepa arm.

Usage

```
data(blaTum)
```

Format

A data.frame contains the following columns:

id patient id

treatment placebo = 0, thiotepa = 1

size size (cm) of largest initial tumor

num initial number of tumors (8 = 8 or more)

time observation time

count number of new tumors since last observation time

Note

To our surprise, the two-treatment (placebo and thiotepa) subset of the full version `bladTumor` do not match the two-treatment version `blaTum`.

References

Byar, D. P. (1980). The Veterans administration study of chemoprophylaxis for recurrent stage I bladder tumors: comparisons of placebo, pyridoxine, and topical thiotepa. *Bladder Tumors and Other Topics in Urological Oncology*, pp. 363–370. New York: Plenum.

Sun, J. and Wei, L. J. (2000) Regression analysis of panel count data with covariate dependent observation and censoring times. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, **62**(2), 293–302.

Huang, C. Y., Wang, M. C. and Zhang, Y. (2006). Analyzing panel count data with informative observation times. *Biometrika*, **93**(4): 763–776.

See Also

`bladTumor`

Examples

```
data(blaTum)
library(ggplot2)
ggplot(blaTum, aes(time, id)) + geom_tile(aes(fill=count)) +
    facet_grid(treatment ~ ., scales="free_y", )
```

`panelReg` *Fits Semiparametric Regression Models for Panel Count Survival Data*

Description

Fits an proportional means model:

$$\Lambda(t; X_i) = E[N_i(t)|X_i] = \Lambda(t)e^{X_i'\beta},$$

where β is a $p \times 1$ vector of covariate coefficient and $\Lambda(\cdot)$ is a completely unspecified baseline mean function.

Estimating procedures include: Wang-Yan's augmented estimating equations (`"AEE"`, `"AEEX"`), Huang-Wang-Zhang's method (`"HWZ"`), Zhang's maximum pseudo-likelihood (`"MPL"`), Maximum pseudolikelihood with I-Splines (`"MPLs"`), Maximum likelihood with I-Splines (`"MLs"`), Sun-Wei's method (`"EE.SWa"`, `"EE.SWb"`, `"EE.SWc"`), and Hu-Sun-Wei's method (`"EE.HSWc"`, `"EE.HSWm"`).

The function can also fits an accelerated mean model (`"AMM"`):

$$\Lambda(t; X_i) = E[N_i(t)|X_i] = \Lambda(te^{X_i'\beta})e^{X_i'\beta}.$$

Usage

```
panelReg(formula, data, method = c("AEE", "AEEX", "HWZ", "MPL", "MPLs",
  "MLs", "AMM", "EE.SWa", "EE.SWb", "EE.SWc", "EE.HSWc", "EE.HSWm"),
  se = c("NULL", "smBootstrap", "Bootstrap", "Impute", "Sandwich"),
  control = list())
```

Arguments

`formula` A formula object, with the response on the left of a "~" operator, and the terms on the right. The response must be a panel count survival object as returned by function `PanelSurv`.

`data` A data.frame in which to interpret the variables named in the `formula`. Three variables including subjects' id, observation times, and number of new events since last observation time are required to feed into function `PanelSurv` as response. At least one covariate variable is required.

`method` Fitting method. See 'Details'.

`se` Standard error estimation method. See 'Details'.

`control` A list of control parameters. See 'Details'.

Details

Some assumptions details about the observation times and censoring time need clarification. Three possible scenarios of observation times are considered: 1) independent observation times – the observation times are independent of the underlying recurrent event process; 2) conditional independent observation times – the observation times are independent of the event process given observed covariates; 3) informative observation times – after conditioning on observed covariates, the observation times and the event process are still dependent through an unobserved multiplicative frailty. Similarly, the three scenarios apply to the censoring time.

`"AEE"` and `"AEEX"` are the augmented estimating equation methods under conditional independent censoring and informative censoring respectively. Both allow informative observation times.

`"HWZ"` is Huang-Wang-Zhang's method. It allows both information observation times and informative censoring time. It does not need to specify the dependence structure or model the frailty.

`"MPL"` and `"MPLs"` are maximum pseudolikelihood methods, with nonparametric and monotone spline estimates of the baseline mean function respectively. They assume conditional independent observation times and censoring time. The underlying event process is assumed to be Poisson, and the within subjects dependence is ignored.

`"MLs"` is maximum likelihood method with monotone splines estimates of the baseline mean function. It assumes conditional independent observation times and censoring time, and a Poisson underlying event process.

`"EE.SWa"`, `"EE.SWb"` and `"EE.SWc"` are estimating equation approaches based on Sun-Wei's methods. The first assumes independent observation times and censoring time. The second assumes conditional independent observation times but independent censoring time. The third assumes conditional independent observation times and censoring time. All three variations work on centered covariates and avoid estimating the baseline mean.

"EE.HSWc" and "EE.HSWm" are estimating equation approaches based on Hu-Sun-Wei's methods. The first assumes independent observation times and censoring time. The second assumes conditional independent observation times but independent censoring time. Both variations work on centered covariates and avoid estimating the baseline mean.

"AMM" is the accelerated mean model. The observation time process is allowed to be correlated with the underlying recurrent event process through a frailty variable, which does not need to be specified. The model also allows marginal interpretations for the regression parameters and connects naturally with AFT model. See Chiou et al. (2017) for more details.

For standard errors estimation method:

"NULL" means do not calculate standard errors; "smBootstrap" is the smoothed bootstrap estimation method that works with "AMM". "Bootstrap" works with all fitting methods; "Impute" is the multiple imputation method that works with "AEE" and "AEEX"; "Sandwich" is the robust sandwich estimation method that works with "AEE" and "AEEX".

The control argument is a list that can supply any of the following components:

betaInit**:** Object of class "numeric", initial value for covariate coefficient, default 0.

interval**:** Object of class "numeric", initial search interval for solving beta, default (-5,5).

maxIter**:** Object of class "numeric", maximum iteration allowed, default 500 for "AEEX" and "HWZ", 150 for others.

absTol**:** Object of class "numeric", absolute tolerance, default 1e-6.

relTol**:** Object of class "numeric", relative tolerance, default 1e-6.

a**:** Object of class "numeric", a tune parameter, default 0.1. In the case of gamma frailty, "a" corresponds to the value of both shape and rate parameters.

R**:** Object of class "numeric", number of bootstrap or imputation, default 30.

Value

An object of S3 class "panelReg" representing the fit. See panelReg.object for details.

References

Chiou, S., Xu, G., Yan, J., and Huang, C.-Y. (2017). Semiparametric estimation of the accelerated mean model with panel count data under informative examination times. *Biometrics*, to appear. <doi: 10.1111/biom.12840>.

Huang, C.-Y., Wang, M., and Zhang, Y. (2006). Analysing panel count data with informative observation times. *Biometrika*, **93**(4), 763–776.

Hu, X. J., Sun, J. and Wei, L. J. (2003). Regression parameter estimation from panel counts. *Scandinavian Journal of Statistics*, **30**, 25–43.

Lu, M., Zhang, Y., and Huang, J. (2007). Estimation of the mean function with panel count data using monotone polynomial splines. *Biometrika*, **94**(3), 705–718.

Sun, J. and Wei, L. J. (2000). Regression analysis of panel count data with covariates-dependent observation and censoring times. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, **62**(2), 293–302.

Wang, X. and Yan, J. (2011). Fitting semiparametric regressions for panel count survival data with an R package spef. *Computer Methods and Programs in Biomedicine*, **104**(2), 278–285.

Wang, X. and Yan, J. (2013). Augmented estimating equations for semiparametric panel count regression with informative observation times and censoring time. *Statistica Sinica*, 359–381.

Zhang, Y. (2002). A Semiparametric pseudolikelihood estimation method for panel count data. *Biometrika*, **89**(1), 39–48.

See Also

panelReg.object

Examples

```
## Not run:
data(blaTum)
## Fit the bladder tumor data set
formula <- PanelSurv(id, time, count) ~ num + size + treatment

panelReg(formula, data = blaTum, method = "AEE", se = "Sandwich")
panelReg(formula, data = blaTum, method = "AEEX", se = "Impute",
         control = list(a = 0.1, R = 30))
panelReg(formula, data = blaTum, method = "HWZ", se = "Bootstrap",
         control = list(R = 30))
panelReg(formula, data = blaTum, method = "MLs", se = "NULL")
panelReg(formula, data = blaTum, method = "EE.SWa", se = "Bootstrap",
         control = list(R = 30))
panelReg(formula, data = blaTum, method = "EE.HSWc", se = "Bootstrap",
         control = list(R = 30))

## End(Not run)
```

panelReg.object *Semiparametric Panel Count Regression Object*

Description

This S3 class of objects is returned by the `panelReg` class of functions to represent a fitted semiparametric panel count regression model. Objects of this class have methods for the functions `print` and `plot`.

Value

beta a vector, estimated coefficients of the linear predictor.

baseline a step function or spline function, estimated cumulative baseline mean.

timeGrid a vector, ordered unique set of observation times.

lambda a vector, estimated baseline mean for each interval.

convergence an integer code, `0` indicates successful convergence, error code 1 indicates that the iteration limit `maxIter` had been reached.

iter an integer value, number of interactions when stopped.

betaSE a vector, estimated standard errors of beta.

betaVar a matrix, estimated covariance matrix of beta.

baselineSE a vector, estimated standard error of cumulative baseline mean.

See Also

`panelReg`

`PanelSurv` *Create a PanelSurv Object*

Description

Create a panel count survival object, usually used as a response variable in a model formula.

Usage

```
PanelSurv(ID, time, count)

is.PanelSurv(x)
```

Arguments

`ID`	Observation subject's ID.
`time`	Observation time.
`count`	Observation subject's ID.
`x`	An `PanelSurv` object.

Value

An object of S3 class `"PanelSurv"`.

psDF a data frame, part of original input data frame with variable "ID", "time" and "count".

timeGrid ordered distinct observation times in the set of all observation times.

panelMatrix a matrix representation of panel count data, one row per subject, one column per time point in `"timeGrid"`.

In the case of `is.PanelSurv`, a logical value `TRUE` if `x` inherits from class `"PanelSurv"`, otherwise an `FALSE`.

In the case of `plot.PanelSurv`, a tile plot of `panelMatrix` produced by package `ggplot2` with color indicating number of counts since last observation time.

See Also

`panelReg`

Examples

```
data(blaTum)
response <- with(blaTum, PanelSurv(id, time, count))
is.PanelSurv(response)
plot(response)
```

plot.panelReg   *Plot a* panelReg *Object.*

Description

Plot the estimated baseline mean function. If "se" option of panelReg is not "NULL", 95 interval is also plotted.

Usage

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

Arguments

x	The result of a call to the panelReg function.
...	Other graphical parameters such as line type, color, or axis labels.

See Also

panelReg panelReg.object

Examples

```
data(blaTum)
## Plot the fit of bladder tumor data set
fm <- PanelSurv(id, time, count) ~ num + size + treatment
fit1 <- panelReg(fm, data=blaTum, method = "AEE", se = "Sandwich")
plot(fit1)

fit2 <- panelReg(fm, data=blaTum, method = "MLs", se = "NULL")
plot(fit2)
```

plot.PanelSurv *Produce Tile Plot*

Description

Plot the tile plot from a `PanelSurv` object.

Usage

```
## S3 method for class 'PanelSurv'
plot(x, heat = FALSE, order = TRUE, ...)
```

Arguments

`x` an object of class `PanelSurv`.

`heat` an optional logical value indicating whether a swimmer-plot-like tile plot will be produced.

`order` an optional logical value indicating whether the tile plot will be sorted by the largest observationt time.

`...` future extension

Value

A `ggplot` object

skinTumor *Skin cancer chemoprevention trial*

Description

A data frame contains data on the recurrence of skin tumor. The original data is available in Table A.3. of Sun and Zhao (2013). This dataset contains records of 290 patients.

Usage

```
data(skinTumor)
```

Format

This data frame contains the following columns:

id : patient id (repeated for each recurrence).

time : observation time.

age : patient's age at enrollment.

male : gender; male = 1, female = 0.

dfmo : treatment (DFMO) group = 1; placebo = 0.

priorTumor : number of prior tumor from diagnosis to randomization.

countBC : number of newly developed basal cell carcinomas tumors since last observation time.

countSC : number of newly developed squamous cell carcinomas tumors since last observation time.

count : number of newly developed non-melanoma tumors since last observation time; this is equal to countBC + countSC.

References

Chiou, S., Xu, G., Yan, J., and Huang, C.-Y. (2017). Semiparametric estimation of the accelerated mean model with panel count data under informative examination times. *Biometrics*, to appear. <doi: 10.1111/biom.12840>.

Sun, J. and Zhao, X. (2013). Statistical Analysis of Panel Count Data. New York: Springer.

See Also

skiTum

Examples

```
data(skinTumor)
library(ggplot2)
ggplot(skinTumor, aes(time, id, width = 25, height = 2)) +
    geom_tile(aes(fill = count)) + theme_bw() +
    facet_grid(dfmo ~ ., scales = "free_y", as.table = FALSE,
               labeller = labeller(dfmo = function(x) paste("DFMO =", x))) +
    scale_fill_gradient(low = "grey", high = "black") +
    scale_x_continuous(breaks = seq(0, 2000, 200)) +
    labs(fill = "Count") + xlab("Time in days")
```

skiTum *A Simulated Data Mimicking a Skin Cancer Chemoprevention Trial*

Description

A data frame contains data on the recurrence of skin tumor. This simulated data is created for illustrative purpose and it mimic the Skin Cancer Chemoprevention Trial used in Chiou et al. (2017). This data set contains records of 100 simulated patients.

Usage

```
data(skiTum)
```

Format

This data frame contains the following columns:

`id` : patient id (repeated for each recurrence).

`time` : observation time.

`age` : patient's age at enrollment; age = 1 if greater than 65, age = 0 otherwise.

`male` : gender; male = 1, female = 0.

`dfmo` : treatment (DFMO) group = 1; placebo = 0.

`priorTumor` : number of prior tumor from diagnosis to randomization.

`count` : number of new tumors since last observation time.

See Also

`skinTumor`

Examples

```
data(skiTum)
library(ggplot2)
skiTum$dfmo <- factor(skiTum$dfmo, levels = c(1, 0), labels = c("placebo", "DFMO"))
ggplot(skiTum, aes(time, id, height = 2, width = 25)) +
    geom_tile(aes(fill = count)) + theme_bw() +
    facet_grid(dfmo ~ ., scales = "free_y") +
    scale_fill_gradient(low = "grey", high = "black") + labs(fill="Count") +
    scale_x_continuous(breaks = seq(0, 1800, 100)) + xlab("Time in days")
```

Index