Package 'discfrail'

October 13, 2022

Title Cox Models for Time-to-Event Data with Nonparametric Discrete
Group-Specific Frailties

Version 0.1

Maintainer Francesca Gasperoni `<francesca.gasperoni@polimi.it>`

Description Functions for fitting Cox proportional hazards models for grouped time-to-
event data, where the shared group-specific frailties have a discrete nonparametric distribu-
tion. The methods proposed in the package is described by Gasper-
oni, F., Ieva, F., Paganoni, A. M., Jack-
son, C. H., Sharples, L. (2018) <doi:10.1093/biostatistics/kxy071>. There are also func-
tions for simulating from these models, with a nonparametric or a parametric baseline haz-
ard function.

URL https://github.com/fgaspe04/discfrail

BugReports https://github.com/fgaspe04/discfrail/issues

Depends R (>= 3.3.0), survival

License GPL-3

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown

Imports Matrix, numDeriv

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Francesca Gasperoni [aut, cre],
Christopher Jackson [aut]

Repository CRAN

Date/Publication 2019-03-13 12:03:29 UTC

R topics documented:

---

discfrail-package   *discfrail: Cox Models for Time-to-Event Data with Nonparametric Discrete Group-Specific Frailties.*

Description

discfrail: Functions for fitting Cox proportional hazards models for grouped time-to-event data, where the shared group-specific frailties have a discrete nonparametric distribution.

Details

The methods are fully described and illustrated in Gasperoni et al. (2018).

Groups of individuals are clustered into a number of latent populations of groups, each with a common frailty that is assumed to act multiplicatively on their hazard. Covariates can be included through proportional hazards, in the manner of the standard Cox model. The baseline hazard is left unspecified as in the Cox model, and estimation is performed by maximum partial likelihood through an EM algorithm.

There are also functions for simulating from these models, and from similar models with a parametric baseline survival function.

[npdf_cox](#) fits nonparametric discrete frailty models. The number of latent populations of groups can either be fixed, or estimated through an automated model comparison and selection procedure.

[plot.npdf](#) plots fitted survival or cumulative hazard curves from the fitted models.

[sim_npdf](#) simulates from a proportional hazards model with an arbitrary baseline survival distribution and discrete shared frailty terms.

[sim_weibdf](#) simulates from a Weibull survival model with discrete shared frailty terms.

Author(s)

Maintainer: Francesca Gasperoni <francesca.gasperoni@polimi.it>

Authors:

- Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

F. Gasperoni, F. Ieva, A.M. Paganoni, C. Jackson, L. Sharples. (2018) Nonparametric frailty Cox models for hierarchical time-to-event data. Biostatistics.

See Also

Useful links:

- <https://github.com/fgaspe04/discfrail>
- Report bugs at <https://github.com/fgaspe04/discfrail/issues>

---

nelsonaalen_npdf	*Nelson-Aalen estimates of group-specific cumulative hazard from a nonparametric discrete frailty model*

Description

Nelson-Aalen estimates of group-specific cumulative hazard from a nonparametric discrete frailty model

Usage

```
nelsonaalen_npdf(x)
```

Arguments

x   A fitted nonparametric discrete frailty model, as returned by [npdf_cox](#) with estK=FALSE.

Value

A list of objects of length equal to the number of groups in the data. Each component is a list, equivalent to the output of [survfit](#) called for the corresponding group with type="fh", but with two additional components:

y0: -log of the survival estimate

sfun0: a step function for y0 in terms of time, estimated using [stepfun](#).

See Also

[plot.nelsonaalen_npdf](#), [plot.npdf](#).

Examples

```
x = npdf_cox( Surv(time, status) ~ x, groups=family, data=weibdata2030, K = 2,
                estK=FALSE, eps_conv=10^-4)
object = nelsonaalen_npdf( x )
```

---

npdf_cox   *Cox model for grouped survival data with nonparametric discrete*
                                shared frailties

Description

This function fits a Cox proportional hazards model to grouped survival data, where the shared group-specific frailties have a nonparametric discrete distribution. An EM algorithm is used to maximise the marginal partial likelihood.

Usage

```
npdf_cox(formula, groups, data, K = 2, estK = TRUE,
  criterion = "BIC", eps_conv = 10^-4, se_method = c("louis",
  "exact"))
```

Arguments

formula	A formula expression in conventional R linear modelling syntax. The response must be a survival time constructed by the [Surv](#) function from the **survival** package, and any covariates are given on the right-hand side. For example,
	`Surv(time, dead) ~ age + sex`
	Only Surv objects of `type="right"` are supported, corresponding to right-censored observations.
groups	name of the variable which indicates the group in which each individual belongs (e.g. the hospital that the individual is treated in). This can be integer, factor or character. The name should be unquoted.
data	A data frame in which to find variables supplied in `formula`. If not given, the variables should be in the working environment.
K	initial number of latent populations, or clusters of groups which have the same discrete frailty.
estK	If `TRUE` (the default) then multiple models are fitted with number of latent groups ranging from 1 to K. The "best fitting" model according to the criterion specified in `criterion` is then highlighted when printing the object returned by this function.
	If `FALSE` then the number of latent populations is fixed at K.

criterion	Criterion used to choose the best-fitting model to highlight when `estK` is `TRUE`.
	`"Laird"` for the Laird criterion. Running from K latent populations to 1 latent population, this criterion selects the maximum number of latent populations that are non empty as the best K.
	`"AIC"` for Akaike's information criterion.
	`"BIC"` for the Bayesian information criterion (the default).
eps_conv	convergence tolerance for the EM algorithm.
se_method	Method or methods used to compute the standard errors. A character vector containing one or more of the following:
	`"louis"` The method of Louis (1982) based on an approximation to the information matrix.
	`"exact"` In this method the standard errors are computed directly from the observed information matrix obtained by analytic differentiation.
	`"numeric"` This method uses numerical differentiation to approximate the information matrix, and is substantially slower.
	By default this is `c("louis","exact")` because these two methods are equally fast. So that SEs from both these two methods are calculated and presented. Set `se_method=NULL` to compute no standard errors.

Value

If `estK=FALSE` this returns a list of class `npdf` which includes information about the model fit, including estimates and standard errors.

If `estK=TRUE` this returns a list of class `npdflist`. This has an element `models` that contains a list of length K, with one component of class `npdf` for each fitted model.

`comparison` is a matrix composed of K rows and 5 columns (K, K_fitted, llik, AIC, BIC). K_fitted is the number of estimated latent populations, which can be equal to or less than K. llik stands for log-likelihood, AIC for Akaike Information Criterion and BIC for Bayesian Information Criterion.

`Kopt` is optimal model under each criterion.

`criterion` is the preferred criterion.

In either case, the data frame used for the fit (the "model frame") is appended as a component `mf`.

References

Gasperoni, F., Ieva, F., Paganoni, A.M., Jackson, C. and Sharples, L. (2018). Nonparametric frailty Cox models for hierarchical time-to-event data. *Biostatistics*.

Laird, N. (1978). Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, 73(364), 805–811.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 44(2), 226–233.

Examples

```
test <- npdf_cox( Surv(time, status) ~ x, groups=family, data=weibdata2030, K = 4, eps_conv=10^-4)
test     # optimal model (by all criteria) has 2 latent populations
test$models[[1]] # examine alternative model with 1 latent population
```

---

plot.nelsonaalen_npdf   *Plot Nelson-Aalen estimates of group-specific cumulative hazard from a nonparametric discrete frailty model*

Description

Plot Nelson-Aalen estimates of group-specific cumulative hazard from a nonparametric discrete frailty model

Usage

```
## S3 method for class 'nelsonaalen_npdf'
plot(x, xlim = NULL, ylim = NULL,
  xlab = NULL, ylab = NULL, cols = NULL, ...)
```

Arguments

x	Object returned by nelsonaalen_npdf representing Nelson-Aalen estimates from a nonparametric discrete frailty model
xlim	x-axis limits (vector of 2)
ylim	x-axis limits (vector of 2)
xlab	x-axis label
ylab	y-axis label
cols	vector of colour names or numbers, of the same length as the number of groups
...	options to pass to the generic plot function

See Also

nelsonaalen_npdf

Examples

```
 x = npdf_cox( Surv(time, status) ~ x, groups=family, data=weibdata2030, K = 2,
                estK=FALSE, eps_conv=10^-4)
object = nelsonaalen_npdf( x )
plot( object )
plot( object, xlim=c(0,200), ylim=c(0,2),
      xlab="Follow-up days", ylab="Nelson-Aalen estimate",
      cols=ifelse(x$belonging==2, "purple", "black"))
```

| plot.npdf | *Survival or cumulative hazard curves from a fitted nonparametric discrete frailty model* |

Description

This function plots estimates of the survival or cumulative hazard for each group, coloured according to the latent population that each group belongs to.

Usage

```
## S3 method for class 'npdf'
plot(x, type = "km", cols = NULL, survfit_opts = NULL,
  na_opts = NULL, ...)
```

Arguments

x	A fitted nonparametric discrete frailty model, as returned by [npdf_cox](#) with estK=FALSE.
type	character. If "km" group-specific Kaplan-Meier estimates of survival are plotted. If "na" group-specific Nelson-Aalen estimates of the cumulative hazard are plotted. The default is "km".
cols	Vector of colour names or numbers, of the same length as the number of groups. If not given, this defaults to x$belonging.
survfit_opts	Optional list of additional arguments to pass to [survfit.formula](#).
na_opts	Optional list of arguments (other than "cols") to pass to [plot.nelsonaalen_npdf](#).
...	Optional arguments to pass to [plot.survfit](#).

Examples

```
result = npdf_cox( Surv(time, status) ~ x, groups = family, data = weibdata2030,
                   K = 2, estK = FALSE, eps_conv=10^-4)
plot( result )
plot( result, type = "km" )
plot( result, cols = ifelse( result$belonging == 1, "purple", "black" ), xlim = c( 0, 150 ) )

## use of survfit_opts.  show only first 10 groups
plot( result, survfit_opts = list(subset = (weibdata2030$family >= 10) ))

plot( result, type = "na" )

## use of na_opts to customise the Nelson-Aalen plot
plot( result, type = "na", cols=ifelse(result$belonging==2, "purple", "black"),
     na_opts = list(xlim=c(0,200), ylim=c(0,2),
                    xlab="Follow-up days",
                    ylab="Nelson-Aalen estimate"))
```

plot.npdflist   *Survival or cumulative hazard curves from a nonparametric discrete*
                                  frailty model chosen by automatic model selection

Description

Survival or cumulative hazard curves from a nonparametric discrete frailty model chosen by automatic model selection

Usage

```
## S3 method for class 'npdflist'
plot(x, K = NULL, ...)
```

Arguments

x   An object returned by `npdf_cox` with estK=TRUE, containing a list of fitted non-
               parametric discrete frailty models

K   The number of latent populations which identifies the specific model to plot
               estimates from. By default this is the model selected by the criterion specified
               when calling `npdf_cox`.

...   Options to customise the plot, passed to `plot.npdf`. See `plot.npdf` for a de-
               scription of these.

See Also

`plot.npdf`

Examples

```
result = npdf_cox( Surv(time, status) ~ x, groups = family, data = weibdata2030,
                   K = 2, eps_conv=10^-4)
plot( result, K = 2 )
plot( result, type = "na" )
plot( result, type = "na", cols=ifelse(result$belonging==2, "purple", "black"),
     na_opts = list(xlim=c(0,200), ylim=c(0,2),
                    xlab="Follow-up days",
                    ylab="Nelson-Aalen estimate"))
```

---

plot.survfit_npdf   *Plot Kaplan-Meier estimates of group-specific cumulative hazard from a nonparametric discrete frailty model*

Description

Plot Kaplan-Meier estimates of group-specific cumulative hazard from a nonparametric discrete frailty model

Usage

```
## S3 method for class 'survfit_npdf'
plot(x, xlab = NULL, ylab = NULL, cols = NULL,
  ...)
```

Arguments

x	Object returned by survfit_npdf representing Kaplan-Meier estimates from a nonparametric discrete frailty model
xlab	x-axis label
ylab	y-axis label
cols	vector of colour names or numbers, of the same length as the number of groups
...	options to pass to the generic plot function

See Also

plot.npdf, survfit_npdf

Examples

```
 x = npdf_cox( Surv(time, status) ~ x, groups=family, data=weibdata2030, K = 2,
                estK=FALSE, eps_conv=10^-4)
object = survfit_npdf( x )
plot( object )
```

---

print.npdf   *Print output from a fitted nonparametric discrete frailty model*

Description

Prints estimates, standard errors, likelihood and model comparison statistics from a fitted nonparametric discrete frailty model

Usage

```
## S3 method for class 'npdf'
print(x, digits = NULL, ...)
```

Arguments

x	A fitted nonparametric discrete frailty model, as returned by [npdf_cox](#) with estK=FALSE
digits	Number of significant digits to present, by default max(1, getOption("digits") - 4)
...	Further options (currently unused)

---

print.npdflist	*Print output from a nonparametric discrete frailty modelling procedure with automatic model selection.*

Description

Prints the model comparison statistics comparing models with different numbers of latent populations, and prints the estimates from the optimal model according to the criterion that was specified when calling [npdf_cox](#) (by default, BIC criterion).

Usage

```
## S3 method for class 'npdflist'
print(x, digits = NULL, ...)
```

Arguments

x	An object returned by [npdf_cox](#) with estK=TRUE, containing a list of fitted nonparametric discrete frailty models
digits	Number of significant digits to present, by default max(1, getOption("digits") - 4)
...	Further options (currently unused)

---

sim_npdf	*Simulation of grouped time-to-event data with nonparametric baseline hazard and discrete shared frailty distribution*

Description

This function returns a dataset generated from a semiparametric proportional hazards model with a shared discrete frailty term, for given cumulative baseline hazard function, hazard ratios, distribution of groups among latent populations, frailty values for each latent population, and randomly-generated covariate values.

Usage

```
sim_npdf(J, N = NULL, beta, Lambda_0_inv, p, w_values, cens_perc)
```

Arguments

J	number of groups in the data
N	number of individuals in each group
beta	vector of log hazard ratios
Lambda_0_inv	inverse cumulative baseline hazard function, that is, with covariate values 0 and frailty ratio 1
p	vector of K elements. The kth element gives the proportion of groups in the kth latent population of groups.
w_values	vector of K distinct frailty values, one for each latent population.
cens_perc	percentage of censored events. Censoring times are assumed to be distributed as a Normal with variance equal to 1.

Value

A data frame with one row for each simulated individual, and the following columns:

family: the group which the individual is in (integers 1, 2, ...)

time: the simulated event time.

status: the simulated survival status. Censoring times are generated from a Normal distribution with standard deviation equal to 1 and the mean is estimated in order to guarantee the determined percentage of censored events. The event time is observed (status=1) if it is less than the censoring time, and censored otherwise (status=0).

x: matrix of covariate values, generated from a standard normal distribution.

belong: the frailty hazard ratio corresponding to the cluster of groups in which the individual's group has been allocated.

References

Wan, F. (2017). Simulating survival data with predefined censoring rates for proportional hazards models. *Statistics in medicine*, 36(5), 838-854.

Examples

```
J <- 100
N <- 40
Lambda_0_inv = function( t, c=0.01, d=4.6 ) ( t^( 1/d ) )/c
beta <- 1.6
p <- c( 0.8, 0.2 )
w_values <- c( 0.8, 1.6 )
cens_perc <- 0.1
data <- sim_npdf( J, N, beta, Lambda_0_inv, p, w_values, cens_perc )
head( data )
```

---

sim_weibdf *Simulation of grouped time-to-event data with Weibull baseline hazard and discrete shared frailty distribution*

Description

This function returns a dataset generated from a Weibull proportional hazards model with a shared discrete frailty term, for given Weibull parameters, hazard ratios, distribution of groups among latent populations, frailty values for each latent population, and randomly-generated covariate values.

Usage

```
sim_weibdf(J, N = NULL, lambda, rho, beta, p, w_values, cens_perc)
```

Arguments

J	number of groups in the data
N	number of individuals in each group
lambda	Weibull baseline rate parameter (see below), interpreted as the rate parameter with covariate values of 0 and frailty ratio 1. For $rho = 1$ this is the baseline hazard.
rho	Weibull shape parameter (see below)
beta	covariate effects in the Weibull distribution, interpreted as log hazard ratios (see below)
p	vector of K elements. The kth element gives the proportion of groups in the kth latent population of groups.
w_values	vector of K distinct frailty values, one for each latent population.
cens_perc	cens_perc percentage of censored events. Censoring times are assumed to be distributed as a Normal with variance equal to 1.

Details

The "proportional hazards" parameterisation of the Weibull distribution is used, with survivor function $S(t) = exp(-\lambda t^\rho w exp(x^T \beta))$. Note this is different from the "accelerated failure time" parameterisation used in, e.g. dweibull. Distribution functions for the proportional hazards parameterisation can be found in the **flexsurv** package.

Value

A data frame with one row for each simulated individual, and the following columns:

`family`: the group which the individual is in (integers 1, 2, ...)

`time`: the simulated event time.

`status`: the simulated survival status. Censoring times are generated from a Normal distribution with standard deviation equal to 1 and the mean is estimated in order to guarantee the determined percentage of censored events. The event time is observed (status=1) if it is less than the censoring time, and censored otherwise (status=0).

`x`: matrix of covariate values, generated from a standard normal distribution.

`belong`: the frailty hazard ratio corresponding to the cluster of groups in which the individual's group has been allocated.

References

Wan, F. (2017). Simulating survival data with predefined censoring rates for proportional hazards models. *Statistics in medicine*, 36(5), 838-854.

Examples

```
J <- 100
N <- 40
lambda <- 0.5
beta <- 1.6
rho <- 1
p <- c( 0.8, 0.2 )
w_values <- c( 0.8, 1.6 )
cens_perc <- 0.2
data <- sim_weibdf( J, N, lambda, rho, beta, p, w_values, cens_perc)
head( data )
```

---

survfit_npdf	*Kaplan-Meier estimates of group-specific cumulative hazard from a nonparametric discrete frailty model*

Description

Kaplan-Meier estimates of group-specific cumulative hazard from a nonparametric discrete frailty model

Usage

```
survfit_npdf(x, survfit_opts = NULL)
```

Arguments

x	A fitted nonparametric discrete frailty model, as returned by `npdf_cox` with `estK=FALSE`.
survfit_opts	Optional list of additional arguments to pass to `survfit.formula`.

Value

A list of survival estimates, one for each group, as produced by `survfit.formula`

See Also

`plot.npdf`, `plot.survfit_npdf`

---

weibdata10040	*Hierarchical time-to-event data simulated from a Weibull baseline distribution with a nonparametric frailty*

Description

Hierarchical time-to-event data simulated from a Weibull baseline distribution with a nonparametric frailty

Usage

```
weibdata10040
```

Format

A data frame with 4000 rows and the following columns

family Integer from 1 to 100 indicating the group membership. There are 40 individuals in each group.

time Survival time

status Indicator for an observed event at the survival time

x Covariate value, generated from a standard normal distribution

belong Underlying group-specific frailty value

There are two underlying latent populations with proportion 0.8 and 0.2 with frailty values 0.8 and 1.6 respectively, thus the second population has twice the hazard of the first. The covariate is associated with a log hazard ratio of 1.6.

The baseline hazard is set to that of a Weibull model.

The percentage of censored events is equal to 20.

The dataset was generated by `sim_weibdf` as follows.

`sim_weibdf( J = 100, N = 40, lambda = 0.3, rho = 1.3, beta = 1.6, p = c( 0.8, 0.2 ), w_values = c( 0.8, 1.6 ), cens_perc = 0.2)`

| weibdata2030 | *Hierarchical time-to-event data simulated from a Weibull baseline distribution with a nonparametric frailty* |

Description

Hierarchical time-to-event data simulated from a Weibull baseline distribution with a nonparametric frailty

Usage

```
weibdata2030
```

Format

A data frame with 600 rows and the following columns

family Integer from 1 to 20 indicating the group membership. Each group has 30 individuals.

time Survival time

status Indicator for an observed event at the survival time

x Covariate value, generated from a standard normal distribution

belong Underlying group-specific frailty value

There are two underlying latent populations with proportion 0.3 and 0.7 with frailty values 0.8 and 1.6 respectively, thus the second population has twice the hazard of the first. The covariate is associated with a log hazard ratio of 1.6.

The baseline hazard is set to that of a Weibull model.

The percentage of censored events is equal to 10.

The dataset was generated by `sim_weibdf` as follows.

```
sim_weibdf( J = 20, N = 30, lambda = 0.6, rho = 1.1, beta = 1.6, p = c( 0.3, 0.7 ), w_values
= c( 0.8, 1.6 ), cens_perc = 0.1)
```

Index