Package 'lnmixsurv'

September 3, 2024

Type Package

Title Bayesian Mixture Log-Normal Survival Model

Version 3.1.6

Date 2024-08-21

Description Bayesian Survival models via the mixture of Log-Normal distribution extends the well-known survival models and accommodates different behaviour over time and considers higher censored survival times. The proposal combines mixture distributions Fruhwirth-Schnatter(2006) <doi:10.1007/s11336-009-9121-4>, and data augmentation techniques Tanner and Wong (1987) <doi:10.1080/01621459.1987.10478458>.

License MIT + file LICENSE

Imports stats, posterior, hardhat (>= 1.3.0), rlang, generics, dplyr, readr, purrr, tidyr, tibble, Rcpp, RcppParallel, tidyselect, broom

LinkingTo Rcpp, RcppArmadillo, RcppGSL, RcppParallel

Depends parsnip (>= 1.1.0), R (>= 4.1.0), survival, ggplot2

Suggests ggsurvfit, plotly, bayesplot, censored, knitr, rmarkdown, tidymodels, testthat (>= 3.0.0), covr, withr, pec

Encoding UTF-8

SystemRequirements C++, GNU, GNU make, GSL

VignetteBuilder knitr

Config/testthat/edition 3

LazyData true

URL `https://vivianalobo.github.io/lnmixsurv/`

RoxygenNote 7.3.2

NeedsCompilation yes

Author Viviana das Graças Ribeiro Lobo [aut],
Thaís Cristina Oliveira da Fonseca [aut],
Mariane Branco Alves [aut],
Vitor Capdeville [aut],
Victor Hugo Soares Ney [cre]

Maintainer Victor Hugo Soares Ney <`victor.s.ney@gmail.com`>

Repository CRAN

Date/Publication 2024-09-03 15:20:08 UTC

Contents

`augment.survival_ln_mixture`
Augment data with information from a survival_ln_mixture object

---

Description

Include information about hazard and survival distribution for each individual in a dataset.

Usage

```
## S3 method for class 'survival_ln_mixture'
augment(x, newdata, eval_time, ...)
```

Arguments

`x`	A `survival_ln_mixture` object.
`newdata`	A `base::data.frame()` or `tibble::tiblle()` containing all the original predictors used to create x.
`eval_time`	a vector with the times where the hazard and survival distribuition will be evaluated.
`...`	Not used.

Value

A `tibble::tibble()` with the original covariates and ther survvial and hazard distributions.

`augment.survival_ln_mixture_em`
Augment data with information from a survival_ln_mixture_em object

---

Description

Include information about hazard and survival distribution for each individual in a dataset.

Usage

```
## S3 method for class 'survival_ln_mixture_em'
augment(x, newdata, eval_time, ...)
```

Arguments

`x`	A `survival_ln_mixture_em` object.
`newdata`	A `base::data.frame()` or `tibble::tiblle()` containing all the original predictors used to create x.
`eval_time`	a vector with the times where the hazard and survival distribuition will be evaluated.
`...`	Not used.

Value

A `tibble::tibble()` with the original covariates and ther survvial and hazard distributions.

`join_empirical_hazard` *Function used to join the empirical hazard to the data*

Description

`join_empirical_hazard()` takes a Kaplan Meier empirical estimate which includes the survival estimates and joins the hazard estimates to it.

Usage

```
join_empirical_hazard(km)
```

Arguments

`km`	Kaplan-Meier estimates, i.e., object generated after running broom::tidy(survfit_obj), in which survfit_obj is a survfit object. Can also be a survfit object.

Value

The same object as inputed, but with the hazard estimates column (hazard_estimate) joined to it.

`nobs.survival_ln_mixture` *Extract the number of observations from* `survival_ln_mixture` *fit.*

Description

Extract the number of observations used in a `survival_ln_mixture` fit.

Usage

```
## S3 method for class 'survival_ln_mixture'
nobs(object, ...)
```

Arguments

`object`	A fitted `survival_ln_mixture` object.
`...`	Not used.

Value

A single integer.

`plot.survival_ln_mixture_em` *Visualizes the path of the EM algorithm*

Description

Visualizes the path of the EM algorithm

Usage

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

Arguments

`x`	A fitted `survival_ln_mixture_em` object.
`...`	Not used.

Value

A `ggplot` object (or `plotly`, if the package is avaiable) of the EM algorithm iterations.

plot_fit_on_data *Function used to quick visualize the fitted values (survival estimate) on the data used to fit the model (via EM algorithm or Gibbs).*

Description

`plot_fit_on_data()` estimates survival/hazard for the data the model was fitted on and plots the results.

Usage

```
plot_fit_on_data(
  model,
  data,
  type = "survival",
  interval = "none",
  level = 0.95
)
```

Arguments

`model`	A `survival_ln_mixture` or `survival_ln_mixture_em` object.
`data`	A `data.frame()` or `tibble()` containing the data used to fit the model. For appropriate behavior, should be the same object used to generate survival_ln_mixture/survival_ln_mixture_e objects.
`type`	A character string specifying the type of plot. The default is "survival", but can be "hazard".
`interval`	A character string specifying the type of interval to be plotted. The default is "none", but can be "credible". The EM algorithm does not provide confidence intervals and this parameter is only support for the Bayesian version (`survival_ln_mixture` object).
`level`	A numeric value between 0 and 1 specifying the level of the confidence interval. The default is 0.95.

Value

A list with two objects, one ggplot (`$ggplot`) with the predictions plotted against the empirical data and a tibble with the predictions (`$preds`).

predict.survival_ln_mixture

Predict from a Lognormal Mixture Model

Description

Predict from a Lognormal Mixture Model

Usage

```
## S3 method for class 'survival_ln_mixture'
predict(
  object,
  new_data,
  type,
  eval_time,
  interval = "none",
  level = 0.95,
  ...
)
```

Arguments

`object`	A `survival_ln_mixture` object.
`new_data`	A data frame or matrix of new predictors.
`type`	A single character. The type of predictions to generate. Valid options are: • `"time"` for the survival time. **not implmeented** • `"survival"` for the survival probability. • `"hazard"` for the hazard.
`eval_time`	For type = "hazard" or type = "survival", the times for the distribution.
`interval`	should interval estimates be added? Options are "none" and "credible".
`level`	the tail area of the intervals. Default value is 0.95.
`...`	Not used, but required for extensibility.

Value

A tibble of predictions. The number of rows in the tibble is guaranteed to be the same as the number of rows in `new_data`.

Note

Categorical predictors must be converted to factors before the fit, otherwise the predictions will fail.

Examples

```
# Categorical variables must be converted to factor before the fit.

require(survival)
# Wrong way of doing
set.seed(1)
mod <- survival_ln_mixture(Surv(time, status == 2) ~ factor(sex), lung, intercept = TRUE)

## Not run:
# this piece of code will throw error
predict(mod, data.frame(sex = 1), type = "survival", eval_time = 100)

## End(Not run)

# Correct way
lung$sex <- factor(lung$sex) # converting to factor before
set.seed(1)
mod2 <- survival_ln_mixture(Surv(time, status == 2) ~ sex, lung, intercept = TRUE)
# Note: the categorical predictors must be a character.
predict(mod2, data.frame(sex = "1"), type = "survival", eval_time = 100)
```

---

predict.survival_ln_mixture_em

Predict from a lognormal_em Mixture Model fitted using EM algorithm.

Description

Predict from a lognormal_em Mixture Model fitted using EM algorithm.

Usage

```
## S3 method for class 'survival_ln_mixture_em'
predict(object, new_data, type, eval_time, ...)
```

Arguments

`object`	A `survival_ln_mixture_em` object.
`new_data`	A data frame or matrix of new predictors.
`type`	A single character. The type of predictions to generate. Valid options are: • `"survival"` for the survival probability. • `"hazard"` for the hazard theoretical hazard.
`eval_time`	For type = "hazard" or type = "survival", the times for the distribution.
`...`	Not used, but required for extensibility.

Value

A tibble of predictions. The number of rows in the tibble is guaranteed to be the same as the number of rows in `new_data`.

Note

Categorical predictors must be converted to factors before the fit, otherwise the predictions will fail.

`simulate_data` *Function to simulate survival data from a mixture of normal distribution.*

Description

`simulate_data()` simulates data from a mixture model.

Usage

```
simulate_data(
  n = 4000,
  mixture_components = 2,
  k = 2,
  percentage_censored = 0.4,
  starting_seed = sample(1:2^28, 1)
)
```

Arguments

`n`	Number of observations desired.
`mixture_components`	Number of mixtures to include in the generation of the data.
`k`	number of covariates generated (the total of covariates will be intercept + (k - 1) covariates).
`percentage_censored`	Percentage of censored observations (defined as decimal value between 0 and 1). This will generate a delta vector in which 1 is an event that ocurred and 0 is a censored observation.
`starting_seed`	Seed to start the random number generation.

Value

A list with two elements: `data` and `real_values`. The `data` element is a tibble with the simulated data. The `real_values` is a tibble with the real values of the parameters used to generate the data.

`sim_data` *Simulated lognormal mixture data.*

Description

A simulated dataset with 10000 observations from a lognormal mixutre model with 2 componentes.

Usage

```
sim_data
```

Format

`sim_data`**:**

A list with two componentes:

- $data: A data frame with 10,000 rows and 3 columns:

  y observed survival time

  delta event indicator. 1 == event, 0 == censored.

  x binary covariate
- $true_vals: A named vector with the true values used to generate the data.

`survival_ln_mixture` *Lognormal mixture model - Gibbs sampler*

Description

`survival_ln_mixture()` fits a Bayesian lognormal mixture model with Gibbs sampling (optional EM algorithm to find local maximum at the likelihood function), as described in LOBO, Viviana GR; FONSECA, Thaís CO; ALVES, Mariane B. Lapse risk modeling in insurance: a Bayesian mixture approach. Annals of Actuarial Science, v. 18, n. 1, p. 126-151, 2024.

Usage

```
survival_ln_mixture(
  formula,
  data,
  intercept = TRUE,
  iter = 1000,
  warmup = floor(iter/10),
  thin = 1,
  chains = 1,
  cores = 1,
  mixture_components = 2,
  show_progress = FALSE,
  em_iter = 0,
  starting_seed = sample(1:2^28, 1),
  use_W = FALSE,
  number_em_search = 200,
  iteration_em_search = 1,
  fast_groups = TRUE,
  ...
)

## Default S3 method:
survival_ln_mixture(formula, ...)

## S3 method for class 'formula'
survival_ln_mixture(formula, data, intercept = TRUE, ...)
```

Arguments

`formula`	A formula specifying the outcome terms on the left-hand side, and the predictor terms on the right-hand side. The outcome must be a survival::Surv object.
`data`	A **data frame** containing both the predictors and the outcome.
`intercept`	A logical. Should an intercept be included in the processed data?
`iter`	A positive integer specifying the number of iterations for each chain (including warmup).
`warmup`	A positive integer specifying the number of warmup (aka burnin) iterations per chain. The number of warmup iterations should be smaller than iter.
`thin`	A positive integer specifying the period for saving samples.
`chains`	A positive integer specifying the number of Markov chains.
`cores`	A positive integer specifying the maximum number of cores to run the chains. Setting this to a value bigger than 1 will automatically trigger the parallel mode
`mixture_components`	number of mixture componentes >= 2.
`show_progress`	Indicates if the code shows the progress of the EM algorithm and the Gibbs Sampler.
`em_iter`	A positive integer specifying the number of iterations for the EM algorithm. The EM algorithm is performed before the Gibbs sampler to find better initial values for the chains. On simulations, values lower than 200 seems to work nice.
`starting_seed`	Starting seed for the sampler. If not specified by the user, uses a random integer between 1 and 2^28 This way we ensure, when the user sets a seed in R, that this is passed into the C++ code.
`use_W`	Specifies is the W (groups weight's matrix for each observation) should be used from EM. It holds W constant through the code, resulting in a faster Bayesian Inference (close to what Empirical Bayes would do). It may helps generating credible intervals for the survival and hazard curves, using the information from the previous EM iteration. Make sure the EM have converged before setting this parameter to true. In doubt, leave this as FALSE, the default.

`number_em_search`	Number of different EM's to search for maximum likelihoods. Recommended to leave, at least, at 100. This value can be set to 0 to disable the search for maximum likelihood initial values.
`iteration_em_search`	Number of iterations for each of the EM's used to find the maximum likelihoods. Recommended to leave at small values, such as from 1 to 5.
`fast_groups`	Use fast computation of groups allocations probabilities, defaults to TRUE. Setting it to FALSE can increase the computation time (a lot) but it's worth trying if the chains are not converging.
`...`	Not currently used, but required for extensibility.

Value

A `survival_ln_mixture` object, which is a list with the following componentes:

`posterior`	A posterior::draws_matrix with the posterior of the parameters of the model.
`nobs`	A integer holding the number of observations used to generate the fit.
`blueprint`	The blueprint component of the output of hardhat::mold

Note

Categorical predictors must be converted to factors before the fit, otherwise the predictions will fail.

Examples

```
# Formula interface
library(survival)
set.seed(1)
mod <- survival_ln_mixture(Surv(time, status == 2) ~ NULL, lung, intercept = TRUE)
```

`survival_ln_mixture_em` *Lognormal mixture model - Expectation-Maximization Algorithm*

Description

`survival_ln_mixture_em()` fits an EM algorithm, as described in LOBO, Viviana GR; FONSECA, Thaís CO; ALVES, Mariane B. Lapse risk modeling in insurance: a Bayesian mixture approach. Annals of Actuarial Science, v. 18, n. 1, p. 126-151, 2024, for modelling mixtures of lognormal distributions applied to survival data.

Usage

```
survival_ln_mixture_em(
  formula,
  data,
  intercept = TRUE,
  iter = 50,
  mixture_components = 2,
  starting_seed = sample(1:2^28, 1),
  number_em_search = 200,
  iteration_em_search = 1,
  show_progress = FALSE,
  ...
)

## Default S3 method:
survival_ln_mixture_em(formula, ...)

## S3 method for class 'formula'
survival_ln_mixture_em(formula, data, intercept = TRUE, ...)
```

Arguments

`formula`	A formula specifying the outcome terms on the left-hand side, and the predictor terms on the right-hand side. The outcome must be a survival::Surv object.
`data`	A **data frame** containing both the predictors and the outcome.
`intercept`	A logical. Should an intercept be included in the processed data?
`iter`	A positive integer specifying the number of iterations for the EM algorithm.
`mixture_components`	number of mixture componentes >= 2.
`starting_seed`	Starting seed for the algorithm. If not specified by the user, uses a random integer between 1 and 2^28 This way we ensure, when the user sets a seed in R, that this is passed into the C++ code.
`number_em_search`	Number of different EM's to search for maximum likelihoods. Recommended to leave, at least, at 100.
`iteration_em_search`	Number of iterations for each of the EM's used to find the maximum likelihoods. Recommended to leave at small values, such as from 1 to 5.
`show_progress`	A logical. Should the progress of the EM algorithm be shown?
`...`	Not currently used, but required for extensibility.

Value

An object of class `survival_ln_mixture_em` containing the following elements:

- `em_iterations`: A data frame containing the EM iterations.

- `nobs`: The number of observations.
- `predictors_name`: The names of the predictors.
- `logLik`: The log-likelihood of the model.
- `mixture_groups`: The number of mixture groups.
- `blueprint`: The blueprint used to process the formula

`tidy.survival_ln_mixture` *Tidying method for a Lognormal Mixture model.*

Description

These method tidy the estimates from `survival_ln_mixture` fits into a summary.

Usage

```
## S3 method for class 'survival_ln_mixture'
tidy(
  x,
  effects = "fixed",
  conf.int = FALSE,
  conf.level = 0.9,
  digits = NULL,
  ...
)
```

Arguments

`x`	Fitted model object (survival_ln_mixture).
`effects`	A character vector including one or more of `"fixed"` and `"auxiliary`.
`conf.int`	If `TRUE` columns for lower (`cred.low`) and upper (`cred.high`) bounds of the posterior uncertainty intervals are included.
`conf.level`	A number between 0 and 1 indicating the desired probability mass to include in the intervals. Only used if `conf.int` = `TRUE`.
`digits`	How many significant digits should be displayed?
`...`	Not used.

Value

A `data.frame` without rownames. When `effects="fixed"` (the default), tidy.survival_ln_mixutre returns one row for each coefficient for each component of the mixture with three columns:

`term`	The name of the corresponding term in the model.
`estimate`	A point estimate of the coefficient (posterior median).

`std.error`	A standard error for the point estimate based on `mad`. See the *Uncertainty estimates* section in `print.stanreg` for more details.

Setting `effects="auxiliary"` will select the precision and proportion of mixture components parameters.

Examples

```
require(survival)
lung$sex <- factor(lung$sex)
set.seed(1)
mod2 <- survival_ln_mixture(Surv(time, status == 2) ~ sex, lung)
tidy(mod2)
tidy(mod2, conf.int = TRUE)
tidy(mod2, effects = c("fixed", "auxiliary"), conf.int = TRUE)
```

`tidy.survival_ln_mixture_em` *Tidying method for a Lognormal Mixture model (fitted via Expectation-Maximization algorithm).*

Description

These method tidy the estimates from `survival_ln_mixture` fits into a short summary. It doesn't contain uncertainty estimates since it's a likelihood maximization algorithm.

Usage

```
## S3 method for class 'survival_ln_mixture_em'
tidy(x, effects = "fixed", digits = NULL, ...)
```

Arguments

`x`	Fitted model object (survival_ln_mixture_em).
`effects`	A character vector including one or more of `"fixed"` and `"auxiliary`.
`digits`	How many significant digits should be displayed?
`...`	Not used.

Value

A `data.frame` without rownames. When `effects="fixed"` (the default), tidy.survival_ln_mixutre returns one row for each coefficient for each component of the mixture with two columns:

`term`	The name of the corresponding term in the model.
`estimate`	A point estimate of the coefficient (last iteration value).

Setting `effects="auxiliary"` will select the precision and proportion of mixture components parameters.

Examples

```
require(survival)
lung$sex <- factor(lung$sex)
set.seed(1)
mod2 <- survival_ln_mixture_em(Surv(time, status == 2) ~ sex, lung)
tidy(mod2)
tidy(mod2, effects = c("fixed", "auxiliary"))
```

Index