Package 'splineCox'

June 12, 2025

Type Package

Title A Two-Stage Estimation Approach to Cox Regression Using M-Spline
Function

Version 0.0.4

Maintainer Ren Teranishi <ren.teranishi1227@gmail.com>

Description Implements a two-stage estimation approach for Cox
regression using five-parameter M-spline functions to model the baseline hazard. It allows
for flexible hazard shapes and model selection based on log-likelihood criteria as described in
Teranishi et al.(2025).
In addition, the package provides functions for constructing and evaluating B-spline copulas
based on five M-spline or I-spline basis functions, allowing users to flexibly model and
compute bivariate dependence structures. Both the copula function and its density can be evalu-
ated.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Imports joint.Cox, ggplot2

Suggests knitr, rmarkdown, spelling

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Ren Teranishi [aut, cre]

Repository CRAN

Date/Publication 2025-06-12 14:00:07 UTC

Contents

---

spline.copula   *B-spline copula using the five M-spline basis functions*

Description

spline.copula computes the B-spline copula (or its density) based on the five-parameter M-spline basis functions. This copula is a specific instance of the B-spline copula family, and can be implemented using matrix-based operations with M.spline and I.spline from the **joint.Cox** package.

Usage

```
spline.copula(u, v, R = "PE1", mat = FALSE, density = FALSE)
```

Arguments

u   A numeric vector of values in [0, 1] (uniform marginals for the first variable).

v   A numeric vector of values in [0, 1] (uniform marginals for the second variable).

R   A 5×5 non-negative coefficient matrix defining the copula structure. The matrix must satisfy the following conditions:

- All entries must be non-negative ($R_{kl} \geq 0$).
- The sum of all entries must be exactly 1.
- The row sums and column sums must equal: c(1/8, 1/4, 1/4, 1/4, 1/8) (in order).

These conditions ensure that the resulting function is a valid copula density. You may also specify one of the built-in presets: "PE1", "PE2",–"PE3", "PN1", "PN2", "PN3", "I", NE1", "NE2", "NE3", "NN1", "NN2", "NN3". Default '"PE1"'.

mat   Logical; if TRUE, returns the full matrix (outer product) of copula evaluations; otherwise returns only the diagonal values, i.e., C(u_i, v_i) or c(u_i, v_i) for i = 1,...,n. Default is FALSE.

density   Logical; if TRUE, evaluates the copula density; if FALSE, evaluates the copula function. Default is FALSE.

Details

If density = TRUE, the function computes the copula **density** $(c(u, v))$; otherwise, it returns the **copula distribution function** $(C(u, v))$. If density = FALSE, it returns the copula function. The implementation uses five M-spline or I-spline basis functions defined on [0,1]. The coefficient matrix is fixed internally but can be modified in advanced use.

Value

A numeric vector or matrix. If mat=FALSE, returns a vector of length length(u); if mat=TRUE, returns a matrix of size length(u) x length(v).

See Also

M.spline, I.spline

Examples

```
# Example data
library(joint.Cox)
library(ggplot2)

N = 50
u = v = seq(from = 0, to = 1, length.out = N)
U = rep(u, N)
V = rep(v, each = N)

c.data = data.frame(U = U, V = V, C = spline.copula(U, V, R = "PE1", density=TRUE))
ggplot(aes(x = U, y = V), data = c.data) +
  geom_contour(aes(x = U, y = V, z = C, colour = after_stat(level)),
  data = c.data, bins=25) + xlab("u") + ylab("v")
```

| splineCox.reg1 | *Fitting the five-parameter spline Cox model giving a specified shape* |

Description

splineCox.reg1 estimates the parameters of a five-parameter spline Cox model based on a specified shape for the baseline hazard function. The function calculates the estimates for the model parameters (beta) and the baseline hazard scale parameter (gamma), using non-linear optimization. If a numeric vector is provided for the model parameter, it will be normalized to have an L1 norm of 1. Additionally, if plot = TRUE, the function generates a plot of the estimated baseline hazard function with 95 The x-axis represents time, and the y-axis represents the estimated hazard. The solid line indicates the estimated hazard function, while the dashed red lines represent the confidence intervals.

Usage

```
splineCox.reg1(
  t.event,
  event,
  Z,
  xi1 = min(t.event),
  xi3 = max(t.event),
  model = "constant",
  p0 = rep(0, 1 + ncol(as.matrix(Z))),
  plot = TRUE
)
```

Arguments

`t.event`	a vector for time-to-event
`event`	a vector for event indicator (=1 event; =0 censoring)
`Z`	a matrix for covariates; nrow(Z)=sample size, ncol(Z)=the number of covariates
`xi1`	lower bound for the hazard function; the default is `min(t.event)`
`xi3`	upper bound for the hazard function; the default is `max(t.event)`
`model`	A character string specifying the shape of the baseline hazard function or a numeric vector of length 5 representing custom weights. If a numeric vector is provided, it will be normalized to have an L1 norm of 1. Available options include: "increase", "constant", "decrease", "unimodal1", "unimodal2", "unimodal3", "bathtub1", "bathtub2", "bathtub3". Default is "constant"
`p0`	Initial values to maximize the likelihood (1 + p parameters; baseline hazard scale parameter and p regression coefficients)
`plot`	A logical value indicating whether to plot the estimated baseline hazard function. If `TRUE`, a plot is generated displaying the estimated baseline hazard function along with its 95 The x-axis represents time, and the y-axis represents the estimated hazard. The solid line indicates the estimated hazard function, while the dashed red lines represent the confidence intervals. Default is `TRUE`.

Value

A list containing the following components:

`model`	A shape of the baseline hazard function or the normalized custom numeric vector used.
`parameter`	A numeric vector of the parameters defining the baseline hazard shape.
`beta`	A named vector with the estimates, standard errors, and 95% confidence intervals for the regression coefficients.
`gamma`	A named vector with the estimate, standard error, and 95% confidence interval for the baseline hazard parameter.
`loglik`	A named vector containing the log-likelihood (`LogLikelihood`), Akaike Information Criterion (`AIC`), and Bayesian Information Criterion (`BIC`).
`plot`	A baseline hazard function plot (if `plot = TRUE`).

References

Teranishi, R.; Furukawa, K.; Emura, T. (2025). A Two-Stage Estimation Approach to Cox Regression Under the Five-Parameter Spline Model Mathematics 13(4), 616. doi:10.3390/math13040616 Available at https://www.mdpi.com/2227-7390/13/4/616

Examples

```
# Example data
library(joint.Cox)
data(dataOvarian)
t.event = dataOvarian$t.event
```

```
event = dataOvarian$event
Z = dataOvarian$CXCL12

reg1 <- splineCox.reg1(t.event, event, Z, model = "constant")
print(reg1)
```

| splineCox.reg2 | *Fitting the five-parameter spline Cox model with a specified shape, selecting the best fit* |

Description

`splineCox.reg2` estimates the parameters of a five-parameter spline Cox model for multiple specified shapes and selects the best-fitting model based on the maximization of the log-likelihood function. This function supports predefined model shapes and custom numeric vectors of length 5. If numeric vectors are provided, they will be normalized to have an L1 norm of 1. Additionally, if `plot = TRUE`, the function generates a plot of the estimated baseline hazard function for the best-fitting model, along with its 95 The x-axis represents time, and the y-axis represents the estimated hazard. The solid line indicates the estimated hazard function, while the dashed red lines represent the confidence intervals.

Usage

```
splineCox.reg2(
  t.event,
  event,
  Z,
  xi1 = min(t.event),
  xi3 = max(t.event),
  model = names(shape.list),
  p0 = rep(0, 1 + ncol(as.matrix(Z))),
  plot = TRUE
)
```

Arguments

`t.event`	a vector for time-to-event
`event`	a vector for event indicator (=1 event; =0 censoring)
`Z`	a matrix for covariates; nrow(Z)=sample size, ncol(Z)=the number of covariates
`xi1`	lower bound for the hazard function; the default is `min(t.event)`
`xi3`	upper bound for the hazard function; the default is `max(t.event)`
`model`	A list of character strings and/or numeric vectors of length 5 specifying the shapes of the baseline hazard function to evaluate. Character options include: "increase", "constant", "decrease", "unimodal1", "unimodal2", "unimodal3", "bathtub1", "bathtub2", "bathtub3". Numeric vectors must be of length 5 and will be normalized to have an L1 norm of 1. Default is `names(shape.list)`, which includes all predefined models.

`p0`	Initial values to maximize the likelihood (1 + p parameters; baseline hazard scale parameter and p regression coefficients)
`plot`	A logical value indicating whether to plot the estimated baseline hazard function. If TRUE, a plot is generated displaying the estimated baseline hazard function along with its 95 The x-axis represents time, and the y-axis represents the estimated hazard. The solid line indicates the estimated hazard function, while the dashed red lines represent the confidence intervals. Default is TRUE.

Value

A list containing the following components:

`model`	A character string indicating the shape of the baseline hazard function used.
`parameter`	A numeric vector of the parameters defining the baseline hazard shape.
`beta`	A named vector with the estimates, standard errors, and 95% confidence intervals for the regression coefficients.
`gamma`	A named vector with the estimate, standard error, and 95% confidence interval for the baseline hazard parameter.
`loglik`	A named vector containing the log-likelihood (`LogLikelihood`), Akaike Information Criterion (`AIC`), and Bayesian Information Criterion (`BIC`) for the best-fitting model.
`other_models`	A data frame containing the log-likelihood (`LogLikelihood`) for all other evaluated models, with model names as row names.
`plot`	A baseline hazard function plot for the best-fitting model (if `plot = TRUE`).

References

Teranishi, R.; Furukawa, K.; Emura, T. (2025). A Two-Stage Estimation Approach to Cox Regression Under the Five-Parameter Spline Model Mathematics 13(4), 616. doi:10.3390/math13040616 Available at https://www.mdpi.com/2227-7390/13/4/616

Examples

```
# Example data
library(joint.Cox)
data(dataOvarian)
t.event = dataOvarian$t.event
event = dataOvarian$event
Z = dataOvarian$CXCL12

M = c("constant", "increase", "decrease")
reg2 <- splineCox.reg2(t.event, event, Z, model = M)
print(reg2)
```

Index