Package 'splineCox'

November 11, 2026

Type Package

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

Version 0.0.9

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 evaluated.
Furthermore, the package supports computation of dependence measures such as Kendall's tau and Spearman's rho, derived analytically from the copula parameters.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Imports joint.Cox, stats

Suggests knitr, rmarkdown, spelling, ggplot2,

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Ren Teranishi [aut, cre]

Repository CRAN

Date/Publication 2026-11-11 11:50:02 UTC

Contents

spline.copula 2
spline.copula.MLE 4

spline.copula.simu 5
splineCox.reg1 6
splineCox.reg2 8

Index **11**

`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,
  Kendall = FALSE,
  Spearman = 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`.

Kendall	Logical; if TRUE, returns Kendall's tau in addition to copula values. Default is FALSE.
Spearman	Logical; if TRUE, returns Spearman's rho in addition to copula values. 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

If both Kendall = FALSE and Spearman = FALSE (default), returns:

- A numeric vector of length length(u) if mat = FALSE.
- A numeric matrix of size length(u) x length(v) if mat = TRUE.

If Kendall = TRUE or Spearman = TRUE, returns a list containing:

- value: A numeric vector or matrix representing the evaluated copula function or copula density.
- Kendall_tau: (Optional) Kendall's tau, included only if Kendall = TRUE.
- Spearman_rho: (Optional) Spearman's rho, included only if Spearman = TRUE.

See Also

M.spline, I.spline

Examples

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

# 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")
```

`spline.copula.MLE` *Maximum likelihood estimation for spline copula parameter matrix*

Description

Performs maximum likelihood estimation of the 5x5 spline copula parameter matrix R using an EM-type algorithm based on M-spline basis functions.

Usage

```
spline.copula.MLE(U, V, E1 = 0.001, E2 = 0.001, R0 = NULL)
```

Arguments

`U`	Numeric vector of length n with values in (0, 1).
`V`	Numeric vector of length n with values in (0, 1).
`E1`	Positive scalar specifying the convergence tolerance for the outer EM iteration updating R.
`E2`	Positive scalar specifying the convergence tolerance for the inner iteration updating auxiliary parameters.
`R0`	Initial values for R. Supply a 5x5 numeric matrix for user-specified initial values, `NULL` for the data-driven initialization, or `"I"` for the independence-copula initialization $R_0 = bb^{\mathsf{T}}$.

Details

The algorithm alternates between updating the spline copula parameter matrix and auxiliary parameters until convergence criteria are satisfied. The independence-copula option uses $b = (1/8, 1/4, 1/4, 1/4, 1/8)^{\mathsf{T}}$ and therefore satisfies the required row- and column-sum constraints at initialization.

Value

A list containing:

R Estimated 5x5 spline copula parameter matrix.

R0 Initial matrix used to start the algorithm.

convergence Matrix recording the iteration number and corresponding log-likelihood values.

Examples

```
n <- 20
R <- matrix(c(1,0,0,0,0,
              0,2,0,0,0,
              0,0,2,0,0,
              0,0,0,2,0,
              0,0,0,0,1)/8, 5, 5, byrow = TRUE)
out <- spline.copula.simu(n, R = R)
```

```
fit <- spline.copula.MLE(out$U, out$V)
fit_independence <- spline.copula.MLE(out$U, out$V, R0 = "I")
```

`spline.copula.simu` *Random generation from the B-spline copula using five M-spline basis functions*

Description

`spline.copula.simu` generates random samples (U, V) from the B-spline copula defined by a 5×5 coefficient matrix `R`. The simulation uses the inverse transform method based on the conditional distribution $V \mid U = u$.

Usage

```
spline.copula.simu(
  n,
  R = "PE1",
  seed = NULL,
  report_tau = TRUE,
  report_rho = TRUE,
  verbose = FALSE
)
```

Arguments

`n`	Integer. Number of samples to generate.
`R`	A 5×5 non-negative coefficient matrix or a preset name: `"PE1"`, `"PE2"`, `"PE3"`, `"PN1"`, `"PN2"`, `"PN3"`, `"I"`, `"NE1"`, `"NE2"`, `"NE3"`, `"NN1"`, `"NN2"`, `"NN3"`.
`seed`	Optional integer for reproducibility.
`report_tau`	Logical. If TRUE, returns empirical and theoretical Kendall's tau.
`report_rho`	Logical. If TRUE, returns empirical and theoretical Spearman's rho.
`verbose`	Logical. If TRUE, prints correlation results.

Details

Given $U \sim \mathrm{Uniform}(0, 1)$, the conditional distribution function of $V \mid U = u$ is

$$F_{V|U=u}(v) = M(u)^\top R I(v),$$

where $M(\cdot)$ and $I(\cdot)$ are the five M-spline and I-spline basis vectors. For each u, a draw V is obtained by solving $F_{V|U=u}(v) = W$ for v, where $W \sim U(0, 1)$.

If `report_tau = TRUE`, the function also returns:

- `tau_emp`: empirical Kendall's tau of the simulated values;

- tau_theory: theoretical Kendall's tau computed using spline.copula with Kendall = TRUE.

If report_rho = TRUE, the function also returns:

- rho_emp: empirical Spearman's rho of the simulated values;
- rho_theory: theoretical Spearman's rho computed using spline.copula with Spearman = TRUE.

Value

A list containing:

- U: simulated U-values;
- V: simulated V-values;
- R: user-specified R (preset name or matrix).

If report_tau = TRUE, the list also returns:

- tau_emp: empirical Kendall's tau;
- tau_theory: theoretical Kendall's tau.

If report_rho = TRUE, the list also returns:

- rho_emp: empirical Spearman's rho;
- rho_theory: theoretical Spearman's rho.

Examples

```
set.seed(123)
out <- spline.copula.simu(2000, R = "PE1",
                          report_tau = TRUE,
                          report_rho = TRUE)
out$tau_emp
out$tau_theory
out$rho_emp
out$rho_theory
```

---

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% confidence intervals. 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% confidence intervals. 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\
- gamma: A named vector with the estimate, standard error, and 95\
- 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% confidence intervals. 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% confidence intervals. 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\
- gamma: A named vector with the estimate, standard error, and 95\
- 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

I.spline, *3*

M.spline, *3*

spline.copula, 2, *6*
spline.copula.MLE, 4
spline.copula.simu, 5
splineCox.reg1, 6
splineCox.reg2, 8