

Package ‘gsscopu’

July 2, 2015

Version 0.9-3

Date 2014-08-24

Title Copula Density and 2-D Hazard Estimation using Smoothing Splines

Author Chong Gu <chong@purdue.edu>

Maintainer Chong Gu <chong@purdue.edu>

Depends R (>= 2.14.0), stats, gss (>= 2.1-0)

Description A collection of routines for the estimation of copula density and 2-D hazard function using smoothing splines.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-07-02 18:31:34

R topics documented:

cdsscopu	2
DiaRet	2
dsscopu	3
hzdrate.sshzd2d	4
sscopu	5
sshzd2d	7
summary.sscopu	9

Index	10
--------------	-----------

cdsscopu	<i>Evaluating 1-D Conditional PDF, CDF, and Quantiles of Copula Density Estimates</i>
----------	---

Description

Evaluate conditional pdf, cdf, and quantiles of copula density estimates.

Usage

```
cdsscopu(object, x, cond, pos=1, int=NULL)
cpsscopu(object, q, cond, pos=1)
cqsscopu(object, p, cond, pos=1)
```

Arguments

object	Object of class "sscopu".
x	Vector of points on which conditional pdf is to be evaluated.
cond	Value of conditioning variables.
pos	Position of variable of interest.
int	Normalizing constant.
q	Vector of points on which conditional cdf is to be evaluated.
p	Vector of probabilities for which conditional quantiles are to be calculated.

Value

A vector of conditional pdf, cdf, or quantiles.

See Also

Fitting functions [sscopu](#) and [sscopu2](#), and [dsscopu](#).

DiaRet	<i>Diabetic Retinopathy</i>
--------	-----------------------------

Description

Time to blindness of 197 diabetic retinopathy patients who received a laser treatment in one eye.

Usage

```
data(DiaRet)
```

Format

A data frame containing 197 observations on the following variables.

id	Patient ID.
time1	Follow-up time of left eye.
time2	Follow-up time of right eye.
status1	Censoring indicator of left eye.
status2	Censoring indicator of right eye.
trt1	Treatment indicator of left eye.
trt2	Treatment indicator of right eye.
type	Type of diabetes.
age	Age of patient at diagnosis.
time.t	Follow-up time of treated eye.
time.u	Follow-up time of untreated eye.
status.t	Censoring indicator of treated eye.
status.u	Censoring indicator of untreated eye.

Source

This is reformatted from the data frame `diabetes` in the R package `timereg` by Thomas H. Scheike.

References

Huster, W.J., Brookmeyer, R., and Self, S.G. (1989), Modelling paired survival data with covariates. *Biometrics*, **45**, 145–56.

dsscopu

Evaluating Copula Density Estimates

Description

Evaluate copula density estimates.

Usage

```
dsscopu(object, x, copu=TRUE)
```

Arguments

object	Object of class "sscopu".
x	Vector or matrix of point(s) on which copula density is to be evaluated.
copu	Flag indicating whether to apply copularization.

Value

A vector of copula density values.

See Also

Fitting functions [sscopu](#) and [sscopu2](#).

`hzdrate.sshzd2d`*Evaluating 2-D Smoothing Spline Hazard Estimates*

Description

Evaluate 2-D smoothing spline hazard estimates by `sshzd2d`.

Usage

```
hzdrate.sshzd2d(object, time, covariates=NULL)
survexp.sshzd2d(object, time, covariates=NULL, job=3)
```

Arguments

<code>object</code>	Object of class "sshzd2d".
<code>time</code>	Matrix or vector of time points on which hazard or survival function is to be evaluated.
<code>covariates</code>	Data frame of covariate values.
<code>job</code>	Flag indicating which survival function to evaluate.

Value

A vector of hazard or survival values.

Note

For `job=1, 2`, `survexp.sshzd2d` returns marginal survival $S_1(t)$ or $S_2(t)$. For `job=3`, `survexp.sshzd2d` returns the 2-D survival $S(t_1, t_2)$.

For `hzdrate.sshzd2d` and `survexp.sshzd2d` with `job=3`, `time` should be a matrix of two columns. For `survexp.sshzd2d` with `job=1, 2`, `time` should be a vector.

When `covariates` is present, its length should be either 1 or that of `time`.

See Also

Fitting function [sshzd2d](#).

Description

Estimate copula densities using tensor-product cubic splines.

Usage

```
sscopu(x, symmetry=FALSE, alpha=1.4, order=NULL, exclude=NULL,
       weights=NULL, id.basis=NULL, nbasis=NULL, seed=NULL,
       qdsz.depth=NULL, prec=1e-7, maxiter=30, skip.iter=dim(x)[2]!=2)
```

```
sscopu2(x, censoring=NULL, truncation=NULL, symmetry=FALSE, alpha=1.4,
        weights=NULL, id.basis=NULL, nbasis=NULL, seed=NULL, prec=1e-7,
        maxiter=30)
```

Arguments

x	Matrix of observations on unit cubes.
symmetry	Flag indicating whether to enforce symmetry, or invariance under coordinate permutation.
order	Highest order of interaction terms in log density. When NULL, it is set to <code>dim(x)[2]</code> internally.
exclude	Pair(s) of marginals whose interactions to be excluded in log density.
alpha	Parameter defining cross-validation score for smoothing parameter selection.
weights	Optional vector of bin-counts for histogram data.
id.basis	Index of observations to be used as "knots."
nbasis	Number of "knots" to be used. Ignored when <code>id.basis</code> is specified.
seed	Seed to be used for the random generation of "knots." Ignored when <code>id.basis</code> is specified.
qdsz.depth	Depth to be used in smolyak.quad for the generation of quadrature.
prec	Precision requirement for internal iterations.
maxiter	Maximum number of iterations allowed for internal iterations.
skip.iter	Flag indicating whether to use initial values of theta and skip theta iteration. See ssanova for notes on skipping theta iteration.
censoring	Optional censoring indicator.
truncation	Optional truncation points.

Details

sscopu is essentially [ssden](#) applied to observations on unit cubes. Instead of variables in data frames, the data are entered as a numerical matrix, and model complexity is globally controlled by the highest order of interactions allowed in log density.

sscopu2 further restricts the domain to the unit square, but allows for possible censoring and truncation. With `censoring==0, 1, 2, 3`, a data point (x_1, x_2) represents exact observation, $[0, x_1]x_2$, $x_1x[0, x_2]$, or $[0, x_1]x[0, x_2]$. With truncation point (t_1, t_2) , the sample is taken from $[0, t_1]x[0, t_2]$ instead of the unit square.

With `symmetry=TRUE`, one may enforce the interchangeability of coordinates so that $f(x_1, x_2) = f(x_2, x_1)$, say.

When $(1, 2)$ is a row in `exclude`, interaction terms involving coordinates 1 and 2 are excluded.

Value

sscopu and sscopu2 return a list object of class "sscopu". [dsscopu](#) can be used to evaluate the estimated copula density. A "copularization" process is applied to the estimated density by default so the resulting marginal densities are guaranteed to be uniform.

[cdsscopu](#), [cpsscopu](#), and [cqsscopu](#) can be used to evaluate 1-D conditional pdf, cdf, and quantiles.

Note

For reasonable execution time in higher dimensions, set `skip.iter=TRUE` in calls to sscopu.

When "Newton iteration diverges" in sscopu, try to use a larger `qdsz.depth`; the default values for dimensions 2, 3, 4, 5, 6+ are 24, 14, 12, 11, 10. To be sure a larger `qdsz.depth` indeed makes difference, verify the cubature size using [smolyak.size](#).

The results may vary from run to run. For consistency, specify `id.basis` or set `seed`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (2013), *Smoothing Spline ANOVA Models*. New York: Springer-Verlag.

Gu, C. (2013), Hazard estimation with bivariate survival data and copula density estimation.

Examples

```
## simulate 2-D data
x <- matrix(runif(200),100,2)
## fit copula density
fit <- sscopu(x)
## "same fit"
fit2 <- sscopu2(x,id=fit$id)
## symmetric fit
fit.s <- sscopu(x,sym=TRUE,id=fit$id)
## Kendall's tau and Spearman's rho
summary(fit); summary(fit2); summary(fit.s)
```

```
## clean up
## Not run: rm(x,fit,fit2,fit.s)
```

sshzd2d

*Estimating 2-D Hazard Function Using Smoothing Splines***Description**

Estimate 2-D hazard function using smoothing spline ANOVA models.

Usage

```
sshzd2d(formula1, formula2, symmetry=FALSE, data, alpha=1.4,
         weights=NULL, subset=NULL, id.basis=NULL, nbasis=NULL, seed=NULL,
         prec=1e-7, maxiter=30, skip.iter=FALSE)

sshzd2d1(formula1, formula2, symmetry=FALSE, data, alpha=1.4,
          weights=NULL, subset=NULL, rho="marginal",
          id.basis=NULL, nbasis=NULL, seed=NULL, prec=1e-7, maxiter=30,
          skip.iter=FALSE)
```

Arguments

formula1	Description of the hazard model to be fit on the first axis.
formula2	Description of the hazard model to be fit on the second axis.
symmetry	Flag indicating whether to enforce symmetry of the two axes.
data	Data frame containing the variables in the model.
alpha	Parameter defining cross-validation scores for smoothing parameter selection.
weights	Optional vector of counts for duplicated data.
subset	Optional vector specifying a subset of observations to be used in the fitting process.
id.basis	Index of observations to be used as "knots."
nbasis	Number of "knots" to be used. Ignored when id.basis is specified.
seed	Seed to be used for the random generation of "knots." Ignored when id.basis is specified.
prec	Precision requirement for internal iterations.
maxiter	Maximum number of iterations allowed for internal iterations.
skip.iter	Flag indicating whether to use initial values of theta and skip theta iteration in marginal hazard estimation.
rho	Choice of rho function for sshzd2d1: "marginal" or "weibull".

Details

The 2-D survival function is expressed as $S(t_1, t_2) = C(S_1(t_1), S_2(t_2))$, where $S_1(t_1)$, $S_2(t_2)$ are marginal survival functions and $C(u_1, u_2)$ is a 2-D copula. The marginal survival functions are estimated via the marginal hazards as in [sshzd](#), and the copula is estimated nonparametrically by calling [sscopu2](#).

When `symmetry=TRUE`, a common marginal survival function $S_1(t)=S_2(t)$ is estimated, and a symmetric copula is estimated such that $C(u_1, u_2) = C(u_2, u_1)$.

Covariates can be incorporated in the marginal hazard models as in [sshzd](#), including parametric terms via `partial` and frailty terms via `random`. Arguments `formula1` and `formula2` are typically model formulas of the same form as the argument `formula` in [sshzd](#), but when `partial` or `random` are needed, `formula1` and `formula2` should be lists with model formulas as the first elements and `partial/random` as named elements; when necessary, variable configurations (that are done via argument type in [sshzd](#)) should also be entered as named elements of lists `formula1/formula2`.

When `symmetry=TRUE`, parallel model formulas must be consistent of each other, such as

```
formula1=list(Surv(t1,d1)~t1*u1,partial=~z1,random=~1|id1)
formula2=list(Surv(t2,d2)~t2*u2,partial=~z2,random=~1|id2)
```

where pairs `t1-t2`, `d1-d2` respectively are different elements in data, pairs `u1-u2`, `z1-z2` respectively may or may not be different elements in data, and factors `id1` and `id2` are typically the same but at least should have the same levels.

Value

`sshzd2d` and `sshzd2d1` return a list object of class "sshzd2d".

`hazardrate.sshzd2d` can be used to evaluate the estimated 2-D hazard function. `survexp.sshzd2d` can be used to calculate estimated survival functions.

Note

`sshzd2d1` executes faster than `sshzd2d`, but often at the cost of performance degradation.

The results may vary from run to run. For consistency, specify `id.basis` or `set.seed`.

Author(s)

Chong Gu, <chong@stat.purdue.edu>

References

Gu, C. (2013), Hazard estimation with bivariate survival data and copula density estimation.

Examples

```
## THE FOLLOWING EXAMPLE IS TIME-CONSUMING
## Not run:
data(DiaRet)
## Common proportional hazard model on the margins
fit <- sshzd2d(Surv(time1,status1)~time1+trt1*type,
```



```

Surv(time2,status2)~time2+trt2*type,
data=DiaRet,symmetry=TRUE)
## Evaluate fitted survival and hazard functions
time <- cbind(c(50,70),c(70,70))
cova <- data.frame(trt1=as.factor(c(1,1)),trt2=as.factor(c(1,0)),
                  type=as.factor(c("juvenile","adult")))
survexp.sshzd2d(fit,time,cov=cova)
hzdrate.sshzd2d(fit,time,cov=cova)
## Association between margins: Kendall's tau and Spearman's rho
summary(fit$copu)
## Clean up
rm(DiaRet,fit,time,cova)
dev.off()

## End(Not run)

```

summary.sscopu	<i>Calculating Kendall's Tau and Spearman's Rho for 2-D Copula Density Estimates</i>
----------------	--

Description

Calculate Kendall's tau and Spearman's rho for 2-D copula density estimates.

Usage

```
## S3 method for class 'sscopu'
summary(object, ...)
```

Arguments

object	Object of class "sscopu".
...	Ignored.

Value

A list containing Kendall's tau and Spearman's rho.

See Also

Fitting functions [sscopu](#) and [sscopu2](#).

Index

*Topic **datasets**

DiaRet, 2

*Topic **distribution**

cdsscopu, 2

dsscopu, 3

sscopu, 5

summary.sscopu, 9

*Topic **models**

cdsscopu, 2

dsscopu, 3

hzdrate.sshzd2d, 4

sscopu, 5

sshzd2d, 7

summary.sscopu, 9

*Topic **smooth**

cdsscopu, 2

dsscopu, 3

hzdrate.sshzd2d, 4

sscopu, 5

sshzd2d, 7

summary.sscopu, 9

*Topic **survival**

hzdrate.sshzd2d, 4

sshzd2d, 7

cdsscopu, 2, 6

cpsscopu, 6

cpsscopu (cdsscopu), 2

cqsscopu, 6

cqsscopu (cdsscopu), 2

DiaRet, 2

dsscopu, 2, 3, 6

hzdrate.sshzd2d, 4, 8

smolyak.quad, 5

smolyak.size, 6

ssanova, 5

sscopu, 2, 3, 5, 9

sscopu2, 2, 3, 8, 9

sscopu2 (sscopu), 5

ssden, 6

sshzd, 8

sshzd2d, 4, 7

sshzd2d1 (sshzd2d), 7

summary.sscopu, 9

survexp.sshzd2d, 8

survexp.sshzd2d (hzdrate.sshzd2d), 4