Package 'SemiParSampleSel'

May 17, 2017

Version 1.5

Author Giampiero Marra, Rosalba Radice, Malgorzata Wojtys and Karol Wyszynski

Maintainer Giampiero Marra <giampiero.marra@ucl.ac.uk>

Title Semi-Parametric Sample Selection Modelling with Continuous or Discrete Response

Description
Routine for fitting continuous or discrete response copula sample selection models with semi-parametric predictors, including linear and nonlinear effects.

Depends R (>= 3.1.1), copula, mgcv, mvtnorm, gamlss.dist

Imports magic, trust, VGAM, Matrix, graphics, grDevices, stats, utils, CDVine, matrixStats

LazyLoad yes

License GPL (>= 2)

URL http://www.ucl.ac.uk/statistics/people/giampieromarra

Repository CRAN

Date/Publication 2017-05-17 17:55:57 UTC

NeedsCompilation no

R topics documented:

`SemiParSampleSel-package`

Semiparametric Sample Selection Modelling with Continuous or Discrete Response

---

Description

`SemiParSampleSel` provides a function for fitting continuous and discrete response (copula) sample selection models with parametric and nonparametric predictor effects. Several bivariate copula distributions are supported. The dependence parameter of the copula distribution as well as the shape and dispersion parameters of the outcome distribution can be specified as functions of semiparametric predictors as well. Smoothness selection is achieved automatically and interval calculations are based on a Bayesian approach.

Details

`SemiParSampleSel` provides a function for flexible sample selection modelling with continuous or discrete response. The underlying representation and estimation of the model is based on a penalized regression spline approach, with automatic smoothness selection. The numerical routine carries out function minimization using a trust region algorithm from the package `trust` in combination with an adaptation of a low level smoothness selection fitting procedure from the package `mgcv`.

`SemiParSampleSel` supports the use of many smoothers as extracted from `mgcv`. Scale invariant tensor product smooths are not currently supported. Estimation is by penalized maximum likelihood with automatic smoothness selection by approximate Un-Biased Risk Estimator (UBRE) score, which can also be viewed as an approximate AIC. The depedence between the selection and outcome equations is modelled through the use of copulas.

Confidence intervals for smooth components and nonlinear functions of the model parameters are derived using a Bayesian approach. Approximate p-values for testing individual smooth terms for equality to the zero function are also provided and based on the approach implemented in mgcv. Functions plot.SemiParSampleSel and summary.SemiParSampleSel extract such information from a fitted SemiParSampleSel object. Model/variable selection is also possible via the use of shrinakge smoothers or information criteria.

Function aver calculates the average outcome corrected for non-random sample selection.

If it makes sense, the dependence parameter of the copula function as well as the shape and dispersion parameters of the outcome distribution can be specified as functions of semiparametric predictors.

Author(s)

Giampiero Marra (University College London, Department of Statistical Science), Rosalba Radice (Birkbeck, University of London, Department of Economics, Mathematics and Statistics), Malgorzata Wojtys (University of Plymouth, School of Computing and Mathematics), Karol Wyszynski (University College London, Department of Statistical Science)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Marra G. and Radice R. (2013), Estimation of a Regression Spline Sample Selection Model. *Computational Statistics and Data Analysis*, 61, 158-173.

Wojtys M., Marra G. and Radice R. (in press), Copula Regression Spline Sample Selection Models: The R Package SemiParSampleSel. *Journal of Statistical Software*.

See Also

SemiParSampleSel

---

aver *Estimated overall average from sample selection model*

Description

aver can be used to calculate the overall estimated average from a sample selection model, with corresponding confidence interval obtained using the delta method.

Usage

```
aver(x, sig.lev = 0.05, sw = NULL, univariate = FALSE, delta = TRUE, n.sim = 100)
```

Arguments

x	A fitted SemiParSampleSel object as produced by SemiParSampleSel().
sig.lev	Significance level.
sw	Survey weights.
univariate	It indicates whether the prevalence is calculated using a (naive/classic) univariate equation model or sample selection model. This option has been introduced to compared adjusted and unadjusted estimates.
delta	If TRUE then the delta method is used for confidence interval calculations, otherwise Bayesian posterior simulation is employed.
n.sim	Number of simulated coefficient vectors from the posterior distribution of the estimated model parameters. This is used when delta = FALSE. It may be increased if more precision is required.

Details

aver estimates the overall average of an outcome of interest when there are missing values that are not at random.

Value

res	It returns three values: lower confidence interval limit, estimated average and upper confidence interval limit.
sig.lev	Significance level used.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

SemiParSampleSel-package, SemiParSampleSel, summary.SemiParSampleSel

Examples

```
## see examples for SemiParSampleSel
```

---

bitsgHs *Internal Function*

Description

It provides the quantites needed to calculate the log-likelihood, gradient and Hessian matrix for penalized or unpenalized maximum likelihood optimization, for a number of copula models.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

conv.check *Some convergence diagnostics*

Description

It takes a fitted SemiParSampleSel object produced by SemiParSampleSel() and produces some diagnostic information about the fitting procedure.

Usage

```
conv.check(x)
```

Arguments

x	A SemiParSampleSel object produced by SemiParSampleSel().

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

SemiParSampleSel

copulaBitsD *Internal Function*

Description

It provides first and second derivatives for copulas with respect to their margins and the association parameter θ.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

fit.SemiParSampleSel *Internal Function*

Description

It performs the optimization using a trust region algorithm as well as automatic smoothing parameter estimation.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

ghss *Internal Function*

Description

It provides the log-likelihood, gradient and Hessian matrix for penalized or unpenalized maximum likelihood optimization, for several bivariate copula distributions and continuous outcome margins.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

ghssD *Internal Function*

Description

It provides the log-likelihood, gradient and Hessian matrix for penalized or unpenalized maximum likelihood optimization, for several bivariate copula distributions and discrete outcome margins.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

ghssDuniv *Internal Function*

Description

It provides the log-likelihood, gradient and Hessian matrix for penalized or unpenalized maximum likelihood optimization of the naive model for discrete outcome margins.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

logLik.SemiParSampleSel *Extract the log likelihood for a fitted SemiParSampleSel*

Description

It extracts the log-likelihood for a fitted `SemiParSampleSel` model.

Usage

```
## S3 method for class 'SemiParSampleSel'
logLik(object, ...)
```

Arguments

`object`	A fitted `SemiParSampleSel` object as produced by `SemiParSampleSel()`.
`...`	Un-used for this function.

Details

Modification of the classic `logLik` which accounts for the estimated degrees of freedom used in `SemiParSampleSel` objects. This function is provided so that information criteria work correctly with `SemiParSampleSel` objects by using the correct degrees of freedom.

Value

Standard `logLik` object.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

AIC, BIC

Examples

```
## see examples for SemiParSampleSel
```

---

marginBitsD *Internal Function*

Description

It provides the first and second derivatives with respect to parameters associated with the margins.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

---

pen *Internal Function*

Description

It provides an overall penalty matrix in a format suitable for estimation conditional on smoothing parameters.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

---

plot.SemiParSampleSel *SemiParSampleSel plotting*

Description

It takes a fitted SemiParSampleSel object produced by SemiParSampleSel() and plots the estimated smooth functions on the scale of the linear predictors.

This function is a wrapper for plot.gam() in mgcv. Please see the documentation of plot.gam() for full details.

Usage

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

Arguments

x	A fitted SemiParSampleSel object as produced by SemiParSampleSel().
eq	The equation from which smooth terms should be considered for printing.
...	Other graphics parameters to pass on to plotting commands, as described for plot.gam in mgcv.

Details

This function produces plots showing the smooth terms of a fitted semiparametric bivariate probit model. In the case of 1-D smooths, the x axis of each plot is labelled using the name of the regressor, while the y axis is labelled as s(regr, edf) where regr is the regressor's name, and edf the effective degrees of freedom of the smooth. For 2-D smooths, perspective plots are produced with the x axes labelled with the first and second variable names and the y axis is labelled as s(var1, var2, edf), which indicates the variables of which the term is a function and the edf for the term.

If seWithMean = TRUE then the intervals include the uncertainty about the overall mean. Note that the smooths are still shown centred. The theoretical arguments and simulation study of Marra and Wood (2012) suggest that seWithMean = TRUE results in intervals with close to nominal frequentist coverage probabilities.

Value

The function generates plots.

WARNING

The function can not deal with smooths of more than 2 variables.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Marra G. and Wood S.N. (2012), Coverage Properties of Confidence Intervals for Generalized Additive Model Components. *Scandinavian Journal of Statistics*, 39(1), 53-74.

See Also

SemiParSampleSel, summary.SemiParSampleSel, predict.SemiParSampleSel

Examples

```
## see examples for SemiParSampleSel
```

post.check *Post-estimation response diagnostics*

Description

It takes the response vector and produces a QQ-plot based on the estimates obtained from the sample selection model. For discrete responses normalized and randomized QQ-plots are produced.

Usage

```
post.check(x, bd = 1)
```

Arguments

x	A fitted SemiParSampleSel object as produced by SemiParSampleSel().
bd	Binomial denominator - maximum value the response can take. This argument is only relevant in the case of BI, BB, ZIBI, ZABI, ZIBB and ZABB.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

predict.SemiParSampleSel *Prediction from fitted SemiParSampleSel model*

Description

It takes a fitted SemiParSampleSel object produced by SemiParSampleSel() and produces predictions for a new set of values of the model covariates or the original values used for the model fit. Standard errors of predictions can be produced and are based on the posterior distribution of the model coefficients.

This function is a wrapper for predict.gam() in mgcv. Please see the documentation of mgcv for full details.

Usage

```
## S3 method for class 'SemiParSampleSel'
predict(object, eq, ...)
```

Arguments

object	A fitted SemiParSampleSel object as produced by SemiParSampleSel().
eq	The equation to be considered for prediction.
...	Other arguments as in predict.gam() in mgcv.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

SemiParSampleSel, aver, plot.SemiParSampleSel, summary.SemiParSampleSel

---

print.aver *Print an aver object*

Description

The print method for a aver object.

Usage

```
## S3 method for class 'aver'
print(x, ...)
```

Arguments

x	A aver object produced by aver().
...	Other arguments.

Details

print.aver prints the lower confidence interval limit, estimated average and upper confidence interval limit.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

aver

print.SemiParSampleSel *Print a SemiParSampleSel object*

Description

The print method for a SemiParSampleSel object.

Usage

```
## S3 method for class 'SemiParSampleSel'
print(x, ...)
```

Arguments

x	A SemiParSampleSel object produced by SemiParSampleSel().
...	Other arguments.

Details

print.SemiParSampleSel prints out the family, model equations, total number of observations, estimated association, spread and shape (if present) coefficients, and total effective degrees of freedom for the penalized or unpenalized model.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

SemiParSampleSel

print.summary.SemiParSampleSel *Print a summary.SemiParSampleSel object*

Description

The print method for a summary.SemiParSampleSel object.

Usage

```
## S3 method for class 'summary.SemiParSampleSel'
print(x, digits = max(3, getOption("digits") - 3),
          signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

x	A summary.SemiParSampleSel object produced by summary.SemiParSampleSel().
digits	Number of digits printed in output.
signif.stars	By default significance stars are printed alongside output.
...	Other arguments.

Details

print.summary.SemiParSampleSel prints model term summaries.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

summary.SemiParSampleSel

---

resp.check *Preliminary response diagnostics*

Description

It takes takes the response vector and produces a histogram and QQ-plot. For discrete responses normalized and randomized QQ-plots are produced.

Usage

```
resp.check(y, margin = "N", bd = 1)
```

Arguments

y	reponse vector
margin	margin of interest. Is the response distributed according to the margin? The default is normal outcome.
bd	Binomial denominator - maximum value the response can take. This argument is only relevant in the case of BI, BB, ZIBI, ZABI, ZIBB and ZABB.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

Examples

```
library(SemiParSampleSel)

# Generating normal response with n = 2000
ys <- rnorm(2000, mean=3, sd=1)
resp.check(ys, margin = "N")

## Not run:

# Generating gamma response with n = 2000
ys <- rgamma(2000, shape=2, rate=3)
resp.check(ys, margin = "G")

# Generating Poisson response with n = 2000
ys <- rPO(2000, mu=3)
resp.check(ys, margin = "P")

# Generating negative binomial response with n = 2000
ys <- rNBI(2000, mu=3, sigma=1)
resp.check(ys, margin = "NB")


## End(Not run)
```

S.m *Internal Function*

Description

It provides penalty matrices in a format suitable for the automatic smoothness estimation procedure.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

SemiParSampleSel *Semiparametric Sample Selection Modelling with Continuous or Discrete Response*

Description

SemiParSampleSel can be used to fit continuous or discrete response sample selection models where the linear predictors are flexibly specified using parametric and regression spline components. The depedence between the selection and outcome equations is modelled through the use of copulas. Regression spline bases are extracted from the package mgcv. Multi-dimensional smooths are available via the use of penalized thin plate regression splines. If it makes sense, the dependence parameter of the chosen bivariate distribution as well as the shape and dispersion parameters of the outcome distribution can be specified as functions of semiparametric predictors.

Usage

```
SemiParSampleSel(formula, data = list(), weights = NULL, subset = NULL,
                   start.v = NULL, start.theta = NULL,
                 BivD = "N", margins = c("probit","N"), fp = FALSE, infl.fac = 1,
                rinit = 1, rmax = 100, iterlimsp = 50, pr.tolsp = 1e-6, bd = NULL,
                   parscale)
```

Arguments

formula	A list of two formulas, one for selection equation and the other for the outcome equation. s terms are used to specify smooth smooth functions of predictors. SemiParSampleSel supports the use shrinkage smoothers for variable selection purposes. See the examples below and the documentation of mgcv for further details on formula specifications. Note that the first formula MUST refer to the selection equation. Furthermore, if it makes sense, more equations can be specified for the other model parameters (see Example 1 below).
data	An optional data frame, list or environment containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which SemiParSampleSel is called.
weights	Optional vector of prior weights to be used in fitting.
subset	Optional vector specifying a subset of observations to be used in the fitting process.
start.v	Starting values for all model parameters can be provided here. Otherwise, these are obtained using an adaptation of the two-stage Heckman sample selection correction approach.
start.theta	A starting value for the association parameter of the copula given in BivD.
BivD	Type of bivariate error distribution employed. Possible choices are "N", "C0", "C90", "C180", "C270", "J0", "J90", "J180", "J270", "G0", "G90", "G180", "G270", "F", "FGM" and "AMH" which stand for bivariate normal, Clayton,

rotated Clayton (90 degrees), survival Clayton, rotated Clayton (270 degrees), Joe, rotated Joe (90 degrees), survival Joe, rotated Joe (270 degrees), Gumbel, rotated Gumbel (90 degrees), survival Gumbel, rotated Gumbel (270 degrees), Frank, Farlie-Gumbel-Morgenstern, and Ali-Mikhail-Haq.

`margins` A two-dimensional vector which specifies the marginal distributions of the selection and outcome equations. The first margin currently admits only "probit" or equivalently "N". The second margin can be "N", "GA", "P", "NB", "D", "PIG", "S", "BB", "BI", "GEOM", "LG", "NBII", "WARING", "YULE", "ZIBB", "ZABB", "ZABI", "ZIBI", "ZALG", "ZANBI", "ZINBI", "ZAP", "ZIP", "ZIP2", "ZIPIG" which stand for normal, gamma, Poisson, negative binomial type I, Delaporte, Poisson inverse Gaussian, Sichel, beta binomial, binomial, geometric, logarithmic, negative binomial type II, Waring, Yule, zero inflated beta binomial, zero altered beta binomial, zero altered binomial, zero inflated binomial, zero altered logarithmic, zero altered negative binomial type I, zero inflated negative binomial type I, zero altered Poisson, zero inflated Poisson, zero inflated Poisson type II and zero inflated Poisson inverse Gaussian.

`fp` If `TRUE`, then a fully parametric model with regression splines if fitted.

`infl.fac` Inflation factor for the model degrees of freedom in the UBRE score. Smoother models can be obtained setting this parameter to a value greater than 1.

`rinit` Starting trust region radius. The trust region radius is adjusted as the algorithm proceeds. See the documentation of `trust` for further details.

`rmax` Maximum allowed trust region radius. This may be set very large. If set small, the algorithm traces a steepest descent path.

`iterlimsp` A positive integer specifying the maximum number of loops to be performed before the smoothing parameter estimation step is terminated.

`pr.tolsp` Tolerance to use in judging convergence of the algorithm when automatic smoothing parameter estimation is used.

`bd` Binomial denominator. To be used in the case of "BB", "BI", "ZIBB", "ZABB", "ZABI", "ZIBI".

`parscale` The algorithm will operate as if optimizing objfun(x / parscale, ...). If missing then no rescaling is done. See the documentation of `trust` for more details.

Details

The association between the responses is modelled by parameter ρ or θ. In a semiparametric bivariate sample selection model the linear predictors are flexibly specified using parametric components and smooth functions of covariates. Replacing the smooth components with their regression spline expressions yields a fully parametric bivariate sample selection model. In principle, classic maximum likelihood estimation can be employed. However, to avoid overfitting, penalized likelihood maximization is used instead. Here the use of penalty matrices allows for the suppression of that part of smooth term complexity which has no support from the data. The tradeoff between smoothness and fitness is controlled by smoothing parameters associated with the penalty matrices. Smoothing parameters are chosen to minimize the approximate Un-Biased Risk Estimator (UBRE) score, which can also be viewed as an approximate AIC.

The optimization problem is solved by a trust region algorithm. Automatic smoothing parameter selection is integrated using a performance-oriented iteration approach (Gu, 1992; Wood, 2004).

Roughly speaking, at each iteration, (i) the penalized weighted least squares problem is solved, and (ii) the smoothing parameters of that problem estimated by approximate UBRE. Steps (i) and (ii) are iterated until convergence. Details of the underlying fitting methods are given in Marra and Radice (2013) and Wojtys et. al (in press).

Value

The function returns an object of class `SemiParSampleSel` as described in `SemiParSampleSelObject`.

WARNINGS

Convergence failure may occur when ρ or θ is very high, and/or the total number and selected number of observations are low, and/or there are important mistakes in the model specification (i.e., using C90 when the model equations are positively associated), and/or there are many smooth components in the model as compared to the number of observations. Convergence failure may also mean that an infinite cycling between steps (i) and (ii) occurs. In this case, the smoothing parameters are set to the values obtained from the non-converged algorithm (`conv.check` will give a warning). In such cases, we recommend re-specifying the model, and/or using some rescaling (see `parscale`).

In the context of non-random sample selection, it would not make much sense to specify the dependence parameter as function of covariates. This is because the assumption is that the dependence parameter models the association between the unobserved confounders in the two equations. However, this option does make sense when it is believed that the association coefficient varies across geographical areas, for instance.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Gu C. (1992), Cross validating non-Gaussian data. *Journal of Computational and Graphical Statistics*, 1(2), 169-179.

Marra G. and Radice R. (2013), Estimation of a Regression Spline Sample Selection Model. *Computational Statistics and Data Analysis*, 61, 158-173.

Wojtys M., Marra G. and Radice R. (in press), Copula Regression Spline Sample Selection Models: The R Package SemiParSampleSel. *Journal of Statistical Software*.

Wood S.N. (2004), Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467), 673-686.

See Also

`aver`, `plot.SemiParSampleSel`, `SemiParSampleSel-package`, `SemiParSampleSelObject`, `conv.check`, `predict.SemiParSampleSel`, `summary.SemiParSampleSel`

Examples

```
library(SemiParSampleSel)

#############################################################################
## Generate data
## Correlation between the two equations and covariate correlation 0.5
## Sample size 2000
#############################################################################

set.seed(0)

n <- 2000

rhC <- rhU <- 0.5

SigmaU <- matrix(c(1, rhU, rhU, 1), 2, 2)
U      <- rmvnorm(n, rep(0,2), SigmaU)

SigmaC <- matrix(rhC, 3, 3); diag(SigmaC) <- 1

cov    <- rmvnorm(n, rep(0,3), SigmaC, method = "svd")
cov    <- pnorm(cov)

bi <- round(cov[,1]); x1 <- cov[,2]; x2 <- cov[,3]

f11 <- function(x) -0.7*(4*x + 2.5*x^2 + 0.7*sin(5*x) + cos(7.5*x))
f12 <- function(x) -0.4*( -0.3 - 1.6*x + sin(5*x))
f21 <- function(x) 0.6*(exp(x) + sin(2.9*x))

ys <-  0.58 + 2.5*bi + f11(x1) + f12(x2) + U[, 1] > 0
y  <- -0.68 - 1.5*bi + f21(x1) +         + U[, 2]
yo <- y*(ys > 0)


dataSim <- data.frame(ys, yo, bi, x1, x2)

## CLASSIC SAMPLE SELECTION MODEL
## the first equation MUST be the selection equation

out <- SemiParSampleSel(list(ys ~ bi + x1 + x2,
                             yo ~ bi + x1),
                        data = dataSim)
conv.check(out)
summary(out)

AIC(out)
BIC(out)
aver(out)


## Not run:
```

```
## SEMIPARAMETRIC SAMPLE SELECTION MODEL

## "cr" cubic regression spline basis      - "cs" shrinkage version of "cr"
## "tp" thin plate regression spline basis - "ts" shrinkage version of "tp"
## for smooths of one variable, "cr/cs" and "tp/ts" achieve similar results
## k is the basis dimension - default is 10
## m is the order of the penalty for the specific term - default is 2

out <- SemiParSampleSel(list(ys ~ bi + s(x1, bs = "tp", k = 10, m = 2) + s(x2),
                             yo ~ bi + s(x1)),
                        data = dataSim)
conv.check(out)
AIC(out)
aver(out)

## compare the two summary outputs
## the second output produces a summary of the results obtained when only
## the outcome equation is fitted, i.e. selection bias is not accounted for

summary(out)
summary(out$gam2)

## estimated smooth function plots
## the red line is the true curve
## the blue line is the naive curve not accounting for selection bias

x1.s <- sort(x1[dataSim$ys>0])
f21.x1 <- f21(x1.s)[order(x1.s)] - mean(f21(x1.s))

plot(out, eq = 2, ylim = c(-1, 0.8)); lines(x1.s, f21.x1, col = "red")
par(new = TRUE)
plot(out$gam2, se = FALSE, col = "blue", ylim = c(-1, 0.8), ylab = "", rug = FALSE)



## SEMIPARAMETRIC SAMPLE SELECTION MODEL with association and dispersion parameters
## depending on covariates as well

out <- SemiParSampleSel(list(ys ~ bi + s(x1) + s(x2),
                             yo ~ bi + s(x1),
                                ~ bi,
                                ~ bi + x1),
                        data = dataSim)
conv.check(out)
summary(out)
out$sigma
out$theta

#
#

####################################################
```

```
## example using Clayton copula with normal margins
#####################################################

set.seed(0)

theta <- 5
sig  <- 1.5

myCop <- archmCopula(family = "clayton", dim = 2, param = theta)

# other copula options are for instance: "amh", "frank", "gumbel", "joe"
# for FGM use the following code:
# myCop <- fgmCopula(theta, dim=2)

bivg  <- mvdc(copula = myCop, c("norm", "norm"),
              list(list(mean = 0, sd = 1),
              list(mean = 0, sd = sig)))
er    <- rMvdc(n, bivg)

ys <-  0.58 + 2.5*bi + f11(x1) + f12(x2) + er[, 1] > 0
y  <- -0.68 - 1.5*bi + f21(x1) +         + er[, 2]
yo <- y*(ys > 0)

dataSim <- data.frame(ys, yo, bi, x1, x2)

out <- SemiParSampleSel(list(ys ~ bi + s(x1) + s(x2),
                             yo ~ bi + s(x1)),
                        data = dataSim, BivD = "C0")
conv.check(out)
summary(out)
aver(out)

x1.s <- sort(x1[dataSim$ys>0])
f21.x1 <- f21(x1.s)[order(x1.s)] - mean(f21(x1.s))

plot(out, eq = 2, ylim = c(-1.1, 1.6)); lines(x1.s, f21.x1, col = "red")
par(new = TRUE)
plot(out$gam2, se = FALSE, col = "blue", ylim = c(-1.1, 1.6), ylab = "", rug = FALSE)

#
#

#########################################################
## example using Gumbel copula with normal-gamma margins
#########################################################

set.seed(0)

k <- 2                                # shape of gamma distribution
miu  <- exp(-0.68 - 1.5*bi + f21(x1)) # mean values of y's (log m = Xb)
lambda <- k/miu                       # rate of gamma distribution

theta <- 6
```

```
# Two-dimensional Gumbel copula with unif margins
gumbel.cop <- onacopula("Gumbel", C(theta, 1:2))

# Random sample from two-dimensional Gumbel copula with uniform margins
U <- rnacopula(n = n, gumbel.cop)

# Margins: normal and gamma
er <- cbind(qnorm(U[,1], 0, 1), qgamma(U[, 2], shape = k, rate = lambda))

ys <- 0.58 + 2.5*bi + f11(x1) + f12(x2) + er[, 1] > 0
y  <- er[, 2]
yo <- y*(ys > 0)

dataSim <- data.frame(ys, yo, bi, x1, x2)

out <- SemiParSampleSel(list(ys ~ bi + s(x1) + s(x2),
                             yo ~ bi + s(x1)),
                        data = dataSim, BivD = "G0", margins = c("N", "G"))
conv.check(out)
summary(out)
aver(out)

x1.s <- sort(x1[dataSim$ys>0])
f21.x1 <- f21(x1.s)[order(x1.s)] - mean(f21(x1.s))

plot(out, eq = 2, ylim = c(-1.1, 1)); lines(x1.s, f21.x1, col = "red")
par(new = TRUE)
plot(out$gam2, se = FALSE, col = "blue", ylim = c(-1.1, 1), ylab = "", rug = FALSE)

#
#


#########################################################
## Example for discrete margins and normal copula
#########################################################


# Creating simulation function
bcds <- function(n, s.tau = 0.2, s.sigma = 1, s.nu = 0.5,
                rhC = 0.2, outcome.margin = "PO", copula = "FGM")  {

# Generating covariates
  SigmaC <- matrix( c(1,rhC,rhC,rhC,rhC,1,rhC,rhC,rhC,rhC,1,rhC,rhC,rhC,rhC,1), 4 , 4)
  covariates    <- rmvnorm(n,rep(0,4),SigmaC, method="svd")
  covariates    <- pnorm(covariates)
  x1 <- covariates[,1]
  x2 <- covariates[,2]
  x3 <- round(covariates[,3])
  x4 <- round(covariates[,4])

  # Establishing copula object
```

```
  if (copula == "FGM") {
      Cop <- fgmCopula(dim = 2, param = iTau(fgmCopula(), s.tau))
    } else if (copula == "N") {
      Cop <- ellipCopula(family = "normal", dim = 2, param = iTau(normalCopula(), s.tau))
    } else if (copula == "AMH") {
      Cop <- archmCopula(family = "amh", dim = 2, param = iTau(amhCopula(), s.tau))
    } else if (copula == "C0") {
     Cop <- archmCopula(family = "clayton", dim = 2, param = iTau(claytonCopula(), s.tau))
    } else if (copula == "F") {
      Cop <- archmCopula(family = "frank", dim = 2, param = iTau(frankCopula(), s.tau))
    } else if (copula == "G0") {
      Cop <- archmCopula(family = "gumbel", dim = 2, param = iTau(gumbelCopula(), s.tau))
    } else if (copula == "J0") {
      Cop <- archmCopula(family = "joe", dim = 2, param = iTau(joeCopula(), s.tau))
    }

  # Setting up equations
  f1 <- function(x) 0.4*(-4 - (5.5*x-2.9) + 3*(4.5*x-2.3)^2 - (4.5*x-2.3)^3)
  f2 <- function(x) x*sin(8*x)
  mu_s  <- 1.0 + f1(x1) - 2.0*x2 + 3.1*x3 - 2.2*x4
  mu_o  <-  exp(1.3 + f2(x1) - 1.9*x2 + 2.4*x3 - 0.1*x4)

  # Creating margin dependent object
  if (outcome.margin == "P") {
        speclist <- list(mu = mu_o)
        outcome.margin2 <- "PO"
     } else if (outcome.margin == "NB") {
        speclist  <- list(mu = mu_o, sigma = s.sigma)
        outcome.margin2 <- "NBI"
     } else if (outcome.margin == "D") {
        speclist  <- list(mu = mu_o, sigma = s.sigma, nu = s.nu)
        outcome.margin2 <- "DEL"
     } else if (outcome.margin == "PIG") {
        speclist  <- list(mu = mu_o, sigma = s.sigma)
        outcome.margin2 <- "PIG"
     } else if (outcome.margin == "S") {
        speclist  <- list(mu = mu_o, sigma = s.sigma, nu = s.nu)
        outcome.margin2 <- "SICHEL"
   }
  spec  <- mvdc(copula = Cop, c("norm", outcome.margin2),
                list(list(mean = mu_s, sd = 1), speclist))

  # Simulating data
  simGen <- rMvdc(n, spec)
  y <- ifelse(simGen[,1]>0, simGen[,2], -99)

  dataSim <- data.frame(y, x1, x2, x3, x4)
  dataSim
}



# Creating plots of the true functional form of x1 in both equations
```

```
xt1  <- seq(0, 1, length.out=200)
xt2  <- seq(0,1, length.out=200)
f1t <- function(x) 0.4*(-4 - (5.5*x-2.9) + 3*(4.5*x-2.3)^2 - (4.5*x-2.3)^3)
f2t <- function(x) x*sin(8*x)
plot(xt1, f1t(xt1))
plot(xt2, f2t(xt2))




# Simulating 1000 deviates

set.seed(0)

dataSim<- bcds(1000, s.tau = 0.6, s.sigma = 0.1, s.nu = 0.5,
                rhC = 0.5, outcome.margin = "NB", copula = "N")
dataSim$y.probit<-ifelse(dataSim$y >= 0, 1, 0)


# Estimating SemiParSampleSel

out1 <- SemiParSampleSel(list(y.probit ~ s(x1) + x2 + x3 + x4, y ~ s(x1) + x2 + x3 + x4),
                          data = dataSim, BivD = "N", margins = c("N", "P"))
conv.check(out1)

out2 <- SemiParSampleSel(list(y.probit ~ s(x1) + x2 + x3 + x4, y ~ s(x1) + x2 + x3 + x4),
                         data = dataSim, BivD = "N", margins = c("N", "NB"))
conv.check(out2)


# Model comparison

AIC(out1)
AIC(out2)
VuongClarke(out1, out2)


# Model diagnostics

summary(out2, cm.plot = TRUE)
plot(out2, eq = 1)
plot(out2, eq = 2)
aver(out2, univariate = TRUE)
aver(out2, univariate = FALSE)

#
#



## End(Not run)
```

SemiParSampleSelObject

Fitted SemiParSampleSel object

Description

A fitted semiparametric sample selection object returned by function SemiParSampleSel and of class "SemiParSampleSel".

Value

fit	List of values and diagnostics extracted from the output of the algorithm. For instance, fit$gradient and fit$S.h return the gradient vector and overall penalty matrix scaled by its smoothing parameters, for the bivariate model. See the documentation of trust for details on the diagnostics provided.
gam1	Univariate fit for selection equation. See the documentation of mgcv for full details.
gam2,gam3,gam4,gam5	Univariate fit for the outcome equation and equations 3, 4 and 5 when present.
gam2.1	Univariate fit for equation 2, estimated using an adaptation of the Heckman sample selection correction procedure.
coefficients	The coefficients of the fitted semiparametric sample selection model.
weights	Prior weights used during model fitting.
sp	Estimated smoothing parameters of the smooth components for the fitted sample selection model.
iter.sp	Number of iterations performed for the smoothing parameter estimation step.
iter.if	Number of iterations performed in the initial step of the algorithm.
iter.inner	Number of iterations performed inside smoothing parameter estimation step.
start.v	Starting values for all model parameters of the semiparametric sample selection algorithm. These are obtained using the Heckman sample selection correction approach when starting values are not provided and the dependence parameter is not specified as a function of a linear predictor.
phi	Estimated dispersion for the response of the outcome equation. In the normal bivariate case, this corresponds to the variance.
sigma	Estimated standard deviation for the response of the outcome equation in the case of normal marginal distribution of the outcome.
shape	Estimated shape parameter for the response of the outcome equation in the case of gamma marginal distribution of the outcome.
nu	Estimated shape parameter for the response of the outcome equation in the case of a discrete distribution.
theta	Estimated coefficient linking the two equations. In the normal bivariate case, this corresponds to the correlation coefficient.

n	Sample size.
n.sel	Number of selected observations.
X1,X2,X3,X4,X5	Design matrices associated with the linear predictors.
X1.d2,X2.d2,X3.d2,X4.d2,X5.d2	Number of columns of the design matrices.
l.sp1,l.sp2,l.sp3,l.sp4,l.sp5	Number of smooth components in the equations.
He	Penalized hessian.
HeSh	Unpenalized hessian.
Vb	Inverse of the penalized hessian. This corresponds to the Bayesian variance-covariance matrix used for 'confidence' interval calculations.
F	This is given by Vb*HeSh.
BivD	Type of bivariate copula distribution employed.
margins	Margins used in the bivariate copula specification.
t.edf	Total degrees of freedom of the estimated sample selection model. It is calculated as sum(diag(F)).
bs.mgfit	A list of values and diagnostics extracted from magic in mgcv.
conv.sp	If TRUE then the smoothing parameter selection algorithm converged.
wor.c	Working model quantities.
eta1,eta2	Estimated linear predictors for the two equations.
y1	Binary outcome of the selection equation.
y2	Dependent variable of the outcome equation.
logLik	Value of the (unpenalized) log-likelihood evaluated at the (penalized or unpenalized) parameter estimates.
fp	If TRUE, then a fully parametric model was fitted.
X2s	Full design matrix of outcome equation.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also

aver, SemiParSampleSel, plot.SemiParSampleSel, predict.SemiParSampleSel, summary.SemiParSampleSel

SOEP *SOEP - German labour mobility data*

Description

The data set orginates from the German Socio-Economic Panel survey of 1984. It contains characteristics of 2651 individuals such as employment, marital status, education and political preferences. Note that the data set is significantly smaller than the original data and any source of identification (e.g., ID number) has been removed.

Usage

```
data(SOEP)
```

References

Wyszynski K. and Marra G. (submitted), Sample selection model for count data: a tutorial in R using the package SemiParSampleSel.

st.theta.star *Internal Function*

Description

It computes a starting value for dependence parameter θ and transforms it depending on the copula employed.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

summary.SemiParSampleSel *SemiParSampleSel summary*

Description

It takes a fitted SemiParSampleSel object produced by SemiParSampleSel() and produces some summaries from it.

Usage

```
## S3 method for class 'SemiParSampleSel'
summary(object, n.sim=1000, s.meth="svd", prob.lev=0.05,
                  cm.plot = FALSE, xlim = c(-3, 3), ylab = "Outcome margin",
                  xlab = "Selection margin", ...)
```

Arguments

object	A fitted SemiParSampleSel object as produced by SemiParSampleSel().
n.sim	The number of simulated coefficient vectors from the posterior distribution of the estimated model parameters. This is used to calculate 'confidence' intervals for θ and ϕ.
s.meth	Matrix decomposition used to determine the matrix root of the covariance matrix. See the documentation of mvtnorm for further details.
prob.lev	Probability of the left and right tails of the posterior distribution used for interval calculations.
cm.plot	If TRUE display contour plot of the model based on average parameter values.
xlim	Maximum and minimum values of the selection margin to be displayed by cm.plot.
ylab	Label of the outcome margin axis.
xlab	Label of the selection margin axis.
...	Other arguments.

Details

Using a low level function in mgcv, based on the results of Marra and Wood (2012), 'Bayesian p-values' are returned for the smooth terms. These have better frequentist performance than their frequentist counterpart. Let $\hat{\mathbf{f}}$ and $\mathbf{V}_f$ denote the vector of values of a smooth term evaluated at the original covariate values and the corresponding Bayesian covariance matrix, and let $\mathbf{V}_f^{r-}$ denote the rank r pseudoinverse of $\mathbf{V}_f$. The statistic used is $T = \hat{\mathbf{f}}'\mathbf{V}_f^{r-}\hat{\mathbf{f}}$. This is compared to a chi-squared distribution with degrees of freedom given by r, which is obtained by biased rounding of the estimated degrees of freedom.

Covariate selection can also be achieved using a single penalty shrinkage approach as shown in Marra and Wood (2011).

See Wojtys et al. (in press) for further details.

Value

`tableP1` Table containing parametric estimates, their standard errors, z-values and p-values for equation 1.

`tableP2,tableP3,tableP4,tableP5`
As above but for equation 2, and equations 3, 4 and 5 if present.

`tableNP1` Table of nonparametric summaries for each smooth component including estimated degrees of freedom, estimated rank, approximate Wald statistic for testing the null hypothesis that the smooth term is zero and corresponding p-value, for equation 1.

`tableNP2,tableNP3,tableNP4,tableNP5`
As above but for equation 2 and equations 3, 4 and 5 if present.

`n` Sample size.

`n.sel` Number of selected observations.

`sigma` Estimated standard deviation for the response of the outcome equation in the case of normal marginal distribution of the outcome.

`shape` Estimated shape parameter for the response of the outcome equation in the case of gamma marginal distribution of the outcome.

`phi` Estimated dispersion for the response of the outcome equation.

`theta` Estimated coefficient linking the two equations.

`nu` Estimated coefficient for the response of the outcome equation when the Delaporte and Sichel distributions are employed.

`formula1,formula2,formula3,formula4,formula5`
Formulas used for equations 1, 2, 3, 4 and 5.

`l.sp1,l.sp2,l.sp3,l.sp4,l.sp5`
Number of smooth components in equations 1, 2, 3, 4 and 5.

`t.edf` Total degrees of freedom of the estimated sample selection model.

`CIsig` 'Confidence' interval for σ in the case of normal marginal distribution of the outcome.

`CIshape` 'Confidence' interval for the shape parameter in the case of gamma distribution of the outcome.

`CInu` 'Confidence' interval for the shape parameter in the case of a discrete distribution of the outcome.

`CIth` 'Confidence' intervals for θ.

`BivD` Selected copula function.

`margins` Margins used in the bivariate copula specification.

`n.sel` Number of selected observations.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Marra G. and Wood S.N. (2011), Practical Variable Selection for Generalized Additive Models. *Computational Statistics and Data Analysis*, 55(7), 2372-2387.

Marra G. and Wood S.N. (2012), Coverage Properties of Confidence Intervals for Generalized Additive Model Components. *Scandinavian Journal of Statistics*, 39(1), 53-74.

Wojtys M., Marra G. and Radice R. (in press), Copula Regression Spline Sample Selection Models: The R Package SemiParSampleSel. *Journal of Statistical Software*.

See Also

SemiParSampleSelObject, plot.SemiParSampleSel, predict.SemiParSampleSel

Examples

```
## see examples for SemiParSampleSel
```

---

theta.tau *Internal Function*

Description

Given an estimated value of θ^*, this function computes θ.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

VuongClarke *Vuong and Clarke tests*

Description

The Vuong and Clarke tests are likelihood-ratio-based tests that can be used for choosing between two non-nested models.

Usage

```
VuongClarke(obj1, obj2, sig.lev = 0.05)
```

Arguments

obj1, obj2	Objects of the two fitted bivariate non-nested models.
sig.lev	Significance level used for testing.

Details

The Vuong (1989) and Clarke (2007) tests are likelihood-ratio-based tests for model selection that use the Kullback-Leibler information criterion. The implemented tests can be used for choosing between two bivariate models which are non-nested.

In the Vuong test, the null hypothesis is that the two models are equally close to the actual model, whereas the alternative is that one model is closer. The test follows asymptotically a standard normal distribution under the null. Assume that the critical region is $(-c, c)$, where c is typically set to 1.96. If the value of the test is higher than c then we reject the null hypothesis that the models are equivalent in favor of the model in obj1. Viceversa if the value is smaller than c. If the value falls in $[-c, c]$ then we cannot discriminate between the two competing models given the data.

In the Clarke test, if the two models are statistically equivalent then the log-likelihood ratios of the observations should be evenly distributed around zero and around half of the ratios should be larger than zero. The test follows asymptotically a binomial distribution with parameters n and 0.5. Critical values can be obtained as shown in Clarke (2007). Intuitively, the model in obj1 is preferred over that in obj2 if the value of the test is significantly larger than its expected value under the null hypothesis $(n/2)$, and vice versa. If the value is not significantly different from $n/2$ then obj1 can be thought of as equivalent to obj2.

Value

It returns two decisions based on the tests and criteria discussed above.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Clarke K. (2007), A Simple Distribution-Free Test for Non-Nested Model Selection. *Political Analysis*, 15, 347-363.

Vuong Q.H. (1989), Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses. *Econometrica*, 57(2), 307-333.

See Also

SemiParSampleSel

working.comp *Internal Function*

Description

It efficiently calculates the working model quantities needed to implement the automatic multiple smoothing parameter estimation procedure by exploiting a result which leads to very fast and stable calculations.

Author(s)

Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

References

Wojtys M., Marra G. and Radice R. (in press), Copula Regression Spline Sample Selection Models: The R Package SemiParSampleSel. *Journal of Statistical Software*.

Index