Package ‘SemiParSampleSel’

February 19, 2015

Version 1.2
Author Giampiero Marra, Rosalba Radice, Malgorzata Wojtys and Karol Wyszynski
Maintainer Giampiero Marra <giampiero.marra@ucl.ac.uk>
Title Semiparametric Sample Selection Modelling with Continuous or Discrete Response
Description Routine for fitting continuous or discrete response copula sample selection models with semiparametric predictors, including linear and nonlinear effects.
Depends R (>= 3.1.1), copula, mgcv, mvtnorm, gamlss.dist
Imports magic, trust, VGAM, Matrix
LazyLoad yes
License GPL (>= 2)
URL http://www.ucl.ac.uk/statistics/people/giampieromarra
Repository CRAN
Date/Publication 2015-01-21 17:14:52
NeedsCompilation no

R topics documented:

SemiParSampleSel-package .................................................. 2
bitsgHs ................................................................. 3
est.aver ................................................................. 3
fit.SemiParSampleSel ..................................................... 4
fit.SemiParSampleSel1 .................................................... 5
ghss ................................................................. 5
ghsSD ................................................................. 5
logLik.SemiParSampleSel .................................................. 6
pen ................................................................. 7
plot.SemiParSampleSel .................................................... 7
predict.SemiParSampleSel ................................................ 8
print.est.aver .......................................................... 9
print.SemiParSampleSel ................................................... 10
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. 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 dependence 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. Functions plot.SemiParSampleSel and summary.SemiParSampleSel extract such information from a fitted SemiParSampleSel object. Model/variable selection is also possible via the use of shrinkage smoothers or information criteria.

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


See Also

SemiParSampleSel

---

**bitsgHs**  
*Internal Function*

**Description**

It provides the quantities needed to calculate the log-likelihood, gradient and Hessian matrix for penalized or unpenalized maximum likelihood optimization, for several copula models.

**Author(s)**

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

**est.aver**  
*Estimated overall average from sample selection model*

**Description**

est.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**

```
est.aver(x, sig.lev = 0.05, sw = NULL, naive = FALSE)```

Arguments

x A fitted SemiParSampleSel object as produced by SemiParSampleSel().
sig.lev Significance level.
sw Survey weights.
naive 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.

Details

est.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

---

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>
fit.SemiParSampleSel

Internal Function

Description

It performs the optimization using a trust region algorithm as well as automatic smoothing parameter estimation. This is more efficient and reliable.

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>

ghss0

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>
logLik.SemiParSampleSel

Extract the log likelihood for a fitted SemiParSampleSel

Description

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

Usage

```r
## 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

```r
## see examples for SemiParSampleSel
```
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 component smooth functions that make it up on the scale of the linear predictor.

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

Usage

```r
## 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 plot showing the smooth terms of a fitted semiparametric bivariate probit model. For plots 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 \( s(\text{regr}, \text{edf}) \) where \( \text{regr} \) is the regressor name, and \( \text{edf} \) the estimated degrees of freedom of the smooth. As 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 \( s(\text{var1}, \text{var2}, \text{edf}) \), which indicates the variables of which the term is a function and the edf for the term.

If `seWithMean = TRUE`, then the confidence intervals include the uncertainty about the overall mean. That is, although each smooth is shown centred, the confidence intervals are obtained as if every
other term in the model was constrained to have average 0 (average taken over the covariate values) except for the smooth being plotted. The theoretical arguments and simulation study of Marra and Wood (2012) suggests 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

See Also
SemParSampleSel, summary.SemParSampleSel, predict.SemParSampleSel

Examples

```r
## see examples for SemParSampleSel

predict.SemParSampleSel

Prediction from fitted SemParSampleSel model
```

Description
It takes a fitted SemParSampleSel object produced by SemParSampleSel() 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. These are based on the posterior distribution of the model coefficients.

Usage

```r
## S3 method for class 'SemParSampleSel'
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, est.aver, plot.SemiParSampleSel, summary.SemiParSampleSel

print.est.aver

Print an est.aver object

Description

The print method for a est.aver object.

Usage

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

Arguments

- **x**: A est.aver object produced by est.aver().
- **...**: Other arguments.

Details

print.est.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

- est.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

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

Arguments

- `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`

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

**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 dependence 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 (isotropic).

**Usage**

```r
SemiParSampleSel(formula, data = list(), weights = NULL, subset = NULL, 
method = "trwl", start.v = NULL, start.theta = NULL, 
BivD = "N", margins = c("N","N"), fp = FALSE, 
infl.fac = 1, pPen1 = NULL, pPen2 = NULL, rinit = 1, 
rmax = 100, iterlimsp = 50, pr.tolsp = 1e-6, 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.
- `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.
- `method`: Estimation method used. It can be either "trwl" or "trwlF". The latter option is much faster and more stable when smoothing parameter estimation is employed, and is the default.
- `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 "N" (normal). The second margin can be "N", "G", "P", "NB", "D", "PIG" or "S" which stand for normal, gamma, Poisson, negative binomial, Delaporte, Poisson inverse Gaussian and Sichel.

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.

pPen1, pPen2
Optional list specifying any penalties to be applied to the parametric model terms of equations 1 and 2.

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.

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 $\rho$ or $\theta$. 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 (submitted).

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

Convergence failure may occur when $\rho$ or $\theta$ 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 (ss.checks will give a warning). In such cases, we recommend re-specifying the model, and/or using some rescaling (see parscale).

Author(s)

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

References


See Also

`est.aver.plot.SemiParSampleSel`, `SemiParSampleSel-package`, `SemiParSampleSelObject`, `ss.checks`, `predict.SemiParSampleSel`, `summary.SemiParSampleSel`

Examples

```r
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)
```
SemiParSampleSel

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)
ss.checks(out)
summary(out)

AIC(out)
BIC(out)
est.aver(out)

## SEMIPARAMETRIC SAMPLE SELECTION MODEL

## "cs" 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)
ss.checks(out)
AIC(out)
est.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)
#
#

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

## Not run:
set.seed(0)
teta <- 10
sig <- 1.5
myCop <- archmCopula(family = "clayton", dim = 2, param = teta)
# other copula options are for instance: "amh", "frank", "gumbel", "joe"
# for FGM use the following code:
# myCop <- fgmcopula(teta, 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]
ys <- 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")
ss.checks(out)
summary(out)
est.aver(out)
x1.s <- sort(x1[dataSim$ys>0])
```r
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

```r
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 = "G8", margins = c("N", "G"))

ss.checks(out)
summary(out)
est.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)

#
#
```
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$gradient` and `fit$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**: Univariate fit for outcome equation.
- **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.
- **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   Design matrix associated with the first linear predictor.
X2   Design matrix associated with the second linear predictor.
X1.d2 Number of columns of the design matrix for equation 1. This is used for internal calculations.
X2.d2 Number of columns of the design matrix for equation 2.
1.sp1 Number of smooth components in equation 1.
1.sp2 Number of smooth components in equation 2.
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 given by the square root of the weight matrix times the pseudo-data vector and design matrix, rw.Z and rw.X.
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.
pPen1, pPen2 List specifying any penalties to be applied to the parametric model terms of the model equations.
X2s   Full design matrix of outcome equation.

Author(s)
Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

See Also
est.aver,SemiParSampleSel.plot.SemiParSampleSel.predict.SemiParSampleSel.summary.SemiParSampleSel
**ss.checks**

Some convergence diagnostics

**Description**

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

**Usage**

```r
ss.checks(x)
```

**Arguments**

- `x`: A `SemiParSampleSel` object produced by `SemiParSampleSel()`.

**Author(s)**

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

**See Also**

- `SemiParSampleSel`

---

**st.theta.star**

Internal Function

**Description**

It computes a starting value for dependence parameter $\theta$ and transforms it depending on the copula used.

**Author(s)**

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

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

Usage

```r
## S3 method for class 'SemiParSampleSel'
summary(object, n.sim = 1000, s.meth = "svd", prob.lev = 0.05, ...)
```

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 $\theta$ and $\phi$.
- `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.
- `...`: Other arguments.

Details

As in the package 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 $\hat{f}$ and $V_f$ denote the vector of values of a smooth term evaluated at the original covariate values and the corresponding Bayesian covariance matrix, and let $V_f^{-\dagger}$ denote the rank $r$ pseudoinverse of $V_f$. The statistic used is $T = \hat{f}' V_f^{-\dagger} \hat{f}$. 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. (submitted) for further details.
Value

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tableP1</td>
<td>Table containing parametric estimates, their standard errors, z-values and p-values for equation 1.</td>
</tr>
<tr>
<td>tableP2</td>
<td>As above but for equation 2.</td>
</tr>
<tr>
<td>tableNP1</td>
<td>Table of nonparametric summaries for each smooth component including estimated degrees of freedom, estimated rank, approximate Wald statistic</td>
</tr>
<tr>
<td></td>
<td>for testing the null hypothesis that the smooth term is zero and corresponding p-value, for equation 1.</td>
</tr>
<tr>
<td>tableNP2</td>
<td>As above but for equation 2.</td>
</tr>
<tr>
<td>n</td>
<td>Sample size.</td>
</tr>
<tr>
<td>n.sel</td>
<td>Number of selected observations.</td>
</tr>
<tr>
<td>sigma</td>
<td>Estimated standard deviation for the response of the outcome equation in the case of normal marginal distribution of the outcome.</td>
</tr>
<tr>
<td>shape</td>
<td>Estimated shape parameter for the response of the outcome equation in the case of gamma marginal distribution of the outcome.</td>
</tr>
<tr>
<td>phi</td>
<td>Estimated dispersion for the response of the outcome equation.</td>
</tr>
<tr>
<td>theta</td>
<td>Estimated coefficient linking the two equations.</td>
</tr>
<tr>
<td>formula1</td>
<td>Formula used for equation 1.</td>
</tr>
<tr>
<td>formula2</td>
<td>Formula used for equation 2.</td>
</tr>
<tr>
<td>l.spl1</td>
<td>Number of smooth components in equation 1.</td>
</tr>
<tr>
<td>l.sp2</td>
<td>Number of smooth components in equation 2.</td>
</tr>
<tr>
<td>t.edf</td>
<td>Total degrees of freedom of the estimated sample selection model.</td>
</tr>
<tr>
<td>CIsig</td>
<td>‘Confidence’ interval for $\sigma$ in the case of normal marginal distribution of the outcome.</td>
</tr>
<tr>
<td>CIshape</td>
<td>‘Confidence’ interval for the shape parameter in the case of gamma distribution of the outcome.</td>
</tr>
<tr>
<td>CInu</td>
<td>‘Confidence’ interval for the shape parameter in the case of a discrete distribution of the outcome.</td>
</tr>
<tr>
<td>CIth</td>
<td>‘Confidence’ intervals for $\theta$.</td>
</tr>
<tr>
<td>BivD</td>
<td>Selected copula function.</td>
</tr>
<tr>
<td>margins</td>
<td>Margins used in the bivariate copula specification.</td>
</tr>
<tr>
<td>n.sel</td>
<td>Number of selected observations.</td>
</tr>
</tbody>
</table>

Author(s)

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


See Also

*SemiParSampleSelObject, plot.SemiParSampleSel, predict.SemiParSampleSel*

Examples

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

---

**theta.tau**

*Internal Function*

**Description**

Given an estimated value of $\theta^*$, this function computes $\theta$ using a proper transformation.

**Author(s)**

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

---

**VuongClarke.bcm**

*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.bcm(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 \(\text{obj1}\). Vice versa 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 \(\text{obj1}\) is preferred over that in \(\text{obj2}\) 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 \(\text{obj1}\) can be thought of as equivalent to \(\text{obj2}\).

For details on the actual implementation of the tests see Radice, Marra and Wojtys (submitted).

Value

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

Author(s)

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

References


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 the band structure of the working linear model.

**Author(s)**  
Maintainer: Giampiero Marra <giampiero.marra@ucl.ac.uk>

**References**  


---

**working.compl**  

**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>
Index

*Topic **AIC**
  logLik.SemiParSampleSel, 6
*Topic **BIC**
  logLik.SemiParSampleSel, 6
*Topic **Clarke test**
  VuongClarke.bcm, 23
*Topic **Vuong test**
  VuongClarke.bcm, 23
*Topic **average outcome**
  est.aver, 3
*Topic **bivariate model**
  VuongClarke.bcm, 23
*Topic **copula**
  SemiParSampleSel, 12
*Topic **diagnostics**
  ss.checks, 20
*Topic **hplot**
  plot.SemiParSampleSel, 7
*Topic **information criteria**
  summary.SemiParSampleSel, 21
*Topic **likelihood ratio test**
  VuongClarke.bcm, 23
*Topic **logLik**
  logLik.SemiParSampleSel, 6
*Topic **model**
  predict.SemiParSampleSel, 8
  summary.SemiParSampleSel, 21
*Topic **non-random sample selection**
  VuongClarke.bcm, 23
*Topic **package**
  SemiParSampleSel-package, 2
*Topic **prediction**
  predict.SemiParSampleSel, 8
*Topic **regression spline**
  SemiParSampleSel, 12
*Topic **regression**
  plot.SemiParSampleSel, 7
  SemiParSampleSel-package, 2
  summary.SemiParSampleSel, 21

*Topic **sample selection model**
  SemiParSampleSel, 12
*Topic **sample selection**
  SemiParSampleSel, 12
  SemiParSampleSel-package, 2
*Topic **semiparametric sample selection modelling**
  est.aver, 3
  print.est.aver, 9
  print.SemiParSampleSel, 10
  print.summary.SemiParSampleSel, 10
  SemiParSampleSel, 12
  SemiParSampleSel-package, 2
  ss.checks, 20
  VuongClarke.bcm, 23
*Topic **shrinkage smoother**
  SemiParSampleSel, 12
  summary.SemiParSampleSel, 21
*Topic **smooth**
  plot.SemiParSampleSel, 7
  SemiParSampleSel, 12
  SemiParSampleSel-package, 2
  summary.SemiParSampleSel, 21
*Topic **variable selection**
  SemiParSampleSel, 12
  SemiParSampleSel-package, 2
  summary.SemiParSampleSel, 21

AIC, 6
BIC, 6
bitsghs, 3
est.aver, 2, 3, 9, 14, 19
fit.SemiParSampleSel, 4
fit.SemiParSampleSel1, 5
ghss, 5
ghssD, 5
logLik, 6
logLik.SemiParSampleSel, 6

pen, 7
plot.SemiParSampleSel, 2, 7, 9, 14, 19, 23
predict.SemiParSampleSel, 8, 8, 14, 19, 23
print.est.aver, 9
print.SemiParSampleSel, 10
print.summary.SemiParSampleSel, 10

S.m, 11
SemiParSampleSel, 2–4, 8–10, 12, 19, 20, 24
SemiParSampleSel-package, 2
SemiParSampleSelObject, 14, 18, 23
ss.checks, 14, 20
st.thetastar, 20
summary.SemiParSampleSel, 2, 4, 8, 9, 11, 14, 19, 21

theta.tau, 23

VuongClarke bcm, 23

working.comp, 25
working.comp1, 25