Package ‘HeckmanEM’

June 2, 2025

Type Package

Title Fit Normal, Student-t or Contaminated Normal Heckman Selection Models

Version 0.2-2

Description It performs maximum likelihood estimation for the Heckman selection model (Normal, Student-t or Contaminated normal) using an EM-algorithm <doi:10.1016/j.jmva.2021.104737>. It also performs influence diagnostic through global and local influence for four possible perturbation schema.

Imports mvtnorm (>= 1.1-0), sampleSelection (>= 1.2-6), MomTrunc (>= 5.79), PerformanceAnalytics (>= 2.0.4), ggplot2, methods

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Marcos Prates [aut, cre, trl] (ORCID: <`https://orcid.org/0000-0001-8077-4898`>),
Victor Lachos [aut] (ORCID: <`https://orcid.org/0000-0002-7239-2459`>),
Dipak Dey [aut],
Marcos Oliveira [aut, ctb],
Christian Galarza [ctb],
Katherine Loor [ctb],
Alejandro Ordonez [ctb]

Maintainer Marcos Prates <`marcosop@est.ufmg.br`>

Repository CRAN

Date/Publication 2025-06-02 11:50:02 UTC

Contents

`CaseDeletion` *Case deletion analysis for Heckman selection model*

Description

This function performs case deletion analysis based on a HeckmanEM object (not available for the contaminated normal model).

Usage

```
CaseDeletion(object)
```

Arguments

`object`	A HeckmanEM object.

Details

This function uses the case deletion approach to study the impact of deleting one or more observations from the dataset on the parameters estimates, using the ideas of Cook (1977) and Zhu et.al. (2001). The GD vector contains the generalized Cook distances

$$\mathrm{GD}_i^1 = \dot{Q}_{[i]}(\widehat{\boldsymbol{\theta}} \mid \widehat{\boldsymbol{\theta}})^\top \left\{-\ddot{Q}(\widehat{\boldsymbol{\theta}} \mid \widehat{\boldsymbol{\theta}})\right\}^{-1} \dot{Q}_{[i]}(\widehat{\boldsymbol{\theta}} \mid \widehat{\boldsymbol{\theta}}),$$

where $\dot{Q}_{[i]}(\widehat{\boldsymbol{\theta}} \mid \widehat{\boldsymbol{\theta}})$ is the gradient vector after dropping the ith observation, and $\ddot{Q}(\widehat{\boldsymbol{\theta}} \mid \widehat{\boldsymbol{\theta}})$ is the Hessian matrix. The benchmark was adapted using the suggestion of Barros et al. (2010). We use $(2 \times \mathrm{npar})/n$ as the benchmark for the GD_i, with npar representing the number of estimated model parameters.

Value

A list of class `HeckmanEM.deletion` with a vector GD of dimension n (see details), and a benchmark value.

References

M. Barros, M. Galea, M. González, V. Leiva, Influence diagnostics in the Tobit censored response model, Statistical Methods & Applications 19 (2010) 379–397.

R. D. Cook, Detection of influential observation in linear regression, Technometrics 19 (1977) 15–18.

H. Zhu, S. Lee, B. Wei, J. Zhou, Case-deletion measures for models with incomplete data, Biometrika 88 (2001) 727–737.

Examples

```
n    <- 100
nu   <- 3
cens <- 0.25

set.seed(13)
w <- cbind(1, runif(n, -1, 1), rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df = nu)

sigma2   <- 1
beta     <- c(1, 0.5)
gamma    <- c(1, 0.3, -.5)
gamma[1] <- -c * sqrt(sigma2)

datas <- rHeckman(x, w, beta, gamma, sigma2, rho = 0.6, nu, family = "T")
y     <- datas$y
cc    <- datas$cc

heckmodel <- HeckmanEM(y, x, w, cc, family = "Normal", iter.max = 50)

global <- CaseDeletion(heckmodel)
plot(global)
```

`HeckmanEM` *Fit the Normal, Student-t or Contaminated normal Heckman Selection model*

Description

‘HeckmanEM()‘ fits the Heckman selection model.

Usage

```
HeckmanEM(
  y,
  x,
  w,
  cc,
  nu = 4,
  family = "T",
  error = 1e-05,
  iter.max = 500,
  im = TRUE,
  criteria = TRUE,
  verbose = TRUE
)
```

Arguments

`y`	A response vector.
`x`	A covariate matrix for the response y.
`w`	A covariate matrix for the missing indicator cc.
`cc`	A missing indicator vector (1=observed, 0=missing) .
`nu`	When using the t- distribution, the initial value for the degrees of freedom. When using the CN distribution, the initial values for the proportion of bad observations and the degree of contamination.
`family`	The family to be used (Normal, T or CN).
`error`	The absolute convergence error for the EM stopping rule.
`iter.max`	The maximum number of iterations for the EM algorithm.
`im`	TRUE/FALSE, boolean to decide if the standard errors of the parameters should be computed.
`criteria`	TRUE/FALSE, boolean to decide if the model selection criteria should be computed.
`verbose`	TRUE/FALSE, boolean to decide if the progress should be printed in the screen.

Value

An object of the class HeckmanEM with all the outputs provided from the function.

Examples

```
n    <- 100
nu   <- 3
cens <- 0.25

set.seed(13)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2   <- 1
beta     <- c(1,0.5)
gamma    <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho = 0.6,nu,family="T")
y <- datas$y
cc <- datas$cc

# Normal EM
res.N <- HeckmanEM(y, x, w, cc, family="Normal",iter.max = 50)
# Student-t EM
res.T <- HeckmanEM(y, x, w, cc, nu = 4, family="T", iter.max = 50)
```

`HeckmanEM.criteria` *Model selection criteria for the Heckman Selection model*

Description

‘HeckmanEM.criteria()‘ calculates the AIC, AICc, BIC selection criteria for the fitted Heckman selection model.

Usage

```
HeckmanEM.criteria(obj)
```

Arguments

`obj`	An object of the class HeckmanEM.

Value

The calculated AIC, AICc, and BIC for the parameters of the fitted model.

Examples

```
n <- 100
family <- "T"
nu <- 4
rho <- .6
cens <- .25

set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2 <- 1

beta <- c(1,0.5)
gamma <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
y <- datas$y
cc <- datas$cc

res <- HeckmanEM(y, x, w, cc, nu = 4, family = "Normal", error = 1e-05, iter.max = 500,
                 im = TRUE, criteria = FALSE)
cr <- HeckmanEM.criteria(res)
```

`HeckmanEM.envelope` *Envelope for the Heckman Selection model*

Description

‘HeckmanEM.envelope()‘ plots the envelope for the fitted Heckman selection model.

Usage

```
HeckmanEM.envelope(obj, envelope = 0.95, ...)
```

Arguments

`obj`	An object of the class HeckmanEM.
`envelope`	The envelope coverage percentage.
`...`	Other option for chart.QQPlot from PerformanceAnalytics package.

Value

A residual plot of the fitted data and its envelope.

Examples

```
n <- 100
family <- "T"
nu <- 4
rho <- .6
cens <- .25

set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2 <- 1

beta <- c(1,0.5)
gamma <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
y <- datas$y
cc <- datas$cc

res <- HeckmanEM(y, x, w, cc, nu = 4, family = "Normal", error = 1e-05, iter.max = 500,
                 im = TRUE, criteria = TRUE)
HeckmanEM.envelope(res, ylab="Normalized Quantile Residuals",xlab="Standard normal quantile",
                   line="quartile", col=c(20,1), pch=19, ylim = c(-5,4))
```

`HeckmanEM.infomat` *Standard error estimation for the Heckman Selection model by the Information Matrix*

Description

'HeckmanEM.infomat()' estimates the standard errors for the parameters for the fitted Heckman selection model.

Usage

```
HeckmanEM.infomat(obj)
```

Arguments

`obj`	An object of the class HeckmanEM.

Value

The estimated standard errors for the parameters of the fitted model.

Examples

```
n <- 100
family <- "T"
nu <- 4
rho <- .6
cens <- .25

set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df=nu)

sigma2 <- 1

beta <- c(1,0.5)
gamma <- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

set.seed(1)
datas <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
y <- datas$y
cc <- datas$cc

res <- HeckmanEM(y, x, w, cc, nu = 4, family = "Normal", error = 1e-05, iter.max = 500,
                 im = FALSE, criteria = TRUE)
im <- HeckmanEM.infomat(res)
```

`Influence` *Influence Analysis for the Heckman Selection model*

Description

This function conducts influence analysis for a given ‘HeckmanEM‘ object. The influence analysis can be conducted using several types of perturbations (not available for the contaminated Normal model).

Usage

```
Influence(object, type, colx = NULL, k = 3.5)
```

Arguments

`object`	A ‘HeckmanEM‘ object to perform the analysis on.
`type`	A character string indicating the type of perturbation to perform. The types can be one of "case-weight","scale","response" and"exploratory".
`colx`	Optional integer specifying the position of the column in the object’s matrix `x` that will undergo perturbation. Only required when type is "exploratory".
`k`	A positive real constant to be used in the benchmark calculation: $M_0 + k \times \mathrm{sd}(M_0)$. Default is 3.5.

Value

Returns a list of class `HeckmanEM.influence` with the following elements:

`M0`	A vector of length n with the aggregated contribution of all eigenvectors of the matrix associated with the normal curvature.
`benchmark`	$M_0 + k \times \mathrm{sd}(M_0)$
`influent`	A vector with the influential observations’ positions.
`type`	The perturbation type.

Author(s)

Marcos Oliveira

References

Insert any relevant references here.

See Also

`HeckmanEM`

Examples

```
n    <- 100
nu   <- 3
cens <- 0.25

set.seed(13)
w <- cbind(1, runif(n, -1, 1), rnorm(n))
x <- cbind(w[,1:2])
c <- qt(cens, df = nu)

sigma2   <- 1
beta     <- c(1, 0.5)
gamma    <- c(1, 0.3, -.5)
gamma[1] <- -c * sqrt(sigma2)

datas <- rHeckman(x, w, beta, gamma, sigma2, rho = 0.6, nu, family = "T")
y     <- datas$y
cc    <- datas$cc

heckmodel <- HeckmanEM(y, x, w, cc, family = "Normal", iter.max = 50)

global <- CaseDeletion(heckmodel)
plot(global)

local_case <- Influence(heckmodel, type = "case-weight")
local_case$influent # influential values here!
plot(local_case)

local_scale <- Influence(heckmodel, type = "scale")
local_scale$influent # influential values here!
plot(local_scale)

local_response <- Influence(heckmodel, type = "response")
local_response$influent # influential values here!
plot(local_response)

local_explore <- Influence(heckmodel, type = "exploratory", colx = 2)
local_explore$influent # influential values here!
plot(local_explore)
```

`rHeckman` *Data generation from the Heckman Selection model (Normal, Student-t or CN)*

Description

‘rHeckman()‘ generates a random sample from the Heckman selection model (Normal, Student-t or CN).

Usage

```
rHeckman(x, w, beta, gamma, sigma2, rho, nu = 4, family = "T")
```

Arguments

`x`	A covariate matrix for the response y.
`w`	A covariate matrix for the missing indicator cc.
`beta`	Values for the beta vector.
`gamma`	Values for the gamma vector.
`sigma2`	Value for the variance.
`rho`	Value for the dependence between the response and missing value.
`nu`	When using the t- distribution, the initial value for the degrees of freedom. When using the CN distribution, the initial values for the proportion of bad observations and the degree of contamination.
`family`	The family to be used (Normal, T, or CN).

Value

Return an object with the response (y) and missing values (cc).

Examples

```
n <- 100
rho <- .6
cens <- 0.25
nu <- 4
set.seed(20200527)
w <- cbind(1,runif(n,-1,1),rnorm(n))
x <- cbind(w[,1:2])

family <- "T"
c <- qt(cens, df=nu)

sigma2 <- 1
beta <- c(1,0.5)
gamma<- c(1,0.3,-.5)
gamma[1] <- -c*sqrt(sigma2)

data <- rHeckman(x,w,beta,gamma,sigma2,rho,nu,family=family)
```

Index