Package 'misclassGLM'

September 4, 2025

Type Package

Title Computation of Generalized Linear Models with Misclassified Covariates Using Side Information

Version 0.3.6

Date 2025-09-03

Maintainer Stephan Dlugosz <stephan.dlugosz@googlemail.com>

Depends R (>= 3.0.0)

Imports stats, Matrix, MASS, ucminf, numDeriv, foreach, mlogit

Suggests parallel

Description Estimates models that extend the standard GLM to take misclassification into account. The models require side information from a secondary data set on the misclassification process, i.e. some sort of misclassification probabilities conditional on some common covariates.
A detailed description of the algorithm can be found in Dlugosz, Mammen and Wilke (2015) <https://ftp.zew.de/pub/zew-docs/dp/dp15043.pdf>.

License GPL-3

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-09-03 22:10:02 UTC

Author Stephan Dlugosz [aut, cre]

Contents

`boot.misclassGLM` *Compute Bootstrapped Standard Errors for* `misclassGLM` *Fits*

Description

Obtain bootstrapped standard errors.

Obtain bootstrapped standard errors.

Usage

```
boot.misclassGLM(ret, Y, X, Pmodel, PX, boot.fraction = 1, repetitions = 1000)
```

```
boot.misclassGLM(ret, Y, X, Pmodel, PX, boot.fraction = 1, repetitions = 1000)
```

Arguments

`ret`	a fitted object of class inheriting from 'misclassGLM'.
`Y`	a vector of integers or numerics. This is the dependent variable.
`X`	a matrix containing the independent variables.
`Pmodel`	a fitted model (e.g. of class 'GLM' or 'mlogit') to implicitly produce variations of the predicted true values probabilities. (Usually conditional on the observed misclassified values and additional covariates.)
`PX`	covariates matrix suitable for predicting probabilities from `Pmodel`, usually including the mismeasured covariate.
`boot.fraction`	fraction of sample to be used for estimating the bootstrapped standard errors, for speedup.
`repetitions`	number of bootstrap samples to be drown.

See Also

`misclassGLM`

`misclassGLM`

`boot.misclassMlogit` *Compute Bootstrapped Standard Errors for* `misclassMlogit` *Fits*

Description

Obtain bootstrapped standard errors.

Usage

```
boot.misclassMlogit(
  ret,
  Y,
  X,
  Pmodel,
  PX,
  boot.fraction = 1,
  repetitions = 1000
)
```

Arguments

`ret`	a fitted object of class inheriting from 'misclassMlogit'.
`Y`	a matrix of 0s and 1s, indicating the target class. This is the dependent variable.
`X`	a matrix containing the independent variables.
`Pmodel`	a fitted model (e.g. of class 'GLM' or 'mlogit') to implicitly produce variations of the predicted true values probabilities. (Usually conditional on the observed misclassified values and additional covariates.)
`PX`	covariates matrix suitable for predicting probabilities from `Pmodel`, usually including the mismeasured covariate.
`boot.fraction`	fraction of sample to be used for estimating the bootstrapped standard errors, for speedup.
`repetitions`	number of bootstrap samples to be drown.

See Also

`misclassMlogit`

mfx.misclassGLM *Compute Marginal Effects for* misclassGLM *Fits*

Description

Obtain marginal Effects.

Obtain marginal Effects.

Usage

```
mfx.misclassGLM(w, x.mean = TRUE, rev.dum = TRUE, digits = 3, ...)

mfx.misclassGLM(w, x.mean = TRUE, rev.dum = TRUE, digits = 3, ...)
```

Arguments

w	a fitted object of class inheriting from 'misclassGLM'.
x.mean	logical, if true computes marginal effects at mean, otherwise average marginal effects.
rev.dum	logical, if true, computes differential effects for switch from 0 to 1.
digits	number of digits to be presented in output.
...	further arguments passed to or from other functions.

See Also

misclassGLM

misclassGLM

mfx.misclassMlogit *Compute Marginal Effects for 'misclassMlogit' Fits*

Description

Obtain marginal effects.

Usage

```
mfx.misclassMlogit(
  w,
  x.mean = TRUE,
  rev.dum = TRUE,
  outcome = 2,
  baseoutcome = 1,
  digits = 3,
  ...
)
```

Arguments

w	a fitted object of class inheriting from 'misclassMlogit'.
x.mean	logical, if true computes marginal effects at mean, otherwise average marginal effects.
rev.dum	logical, if true, computes differential effects for switch from 0 to 1.
outcome	for which the ME should be computed.
baseoutcome	base outcome, e.g. reference class of the model.
digits	number of digits to be presented in output.
...	further arguments passed to or from other functions.

See Also

misclassMlogit

---

misclassGLM *GLM estimation under misclassified covariate*

Description

misclassGLM computes estimator for a GLM with a misclassified covariate using additional side information on the misclassification process

misclassGLM computes estimator for a GLM with a misclassified covariate using additional side information on the misclassification process

Usage

```
misclassGLM(
  Y,
  X,
  setM,
  P,
  na.action = na.omit,
  family = gaussian(link = "identity"),
  control = list(),
  par = NULL,
  x = FALSE,
  robust = FALSE
)

misclassGLM(
  Y,
  X,
  setM,
  P,
```

```
  na.action = na.omit,
  family = gaussian(link = "identity"),
  control = list(),
  par = NULL,
  x = FALSE,
  robust = FALSE
)
```

Arguments

Y	a vector of integers or numerics. This is the dependent variable.
X	a matrix containing the independent variables.
setM	(optional) matrix, rows containing potential patterns for a misclassified (latent) covariate M in any coding for a categorical independent variable, e.g. dummy coding (default: Identity).
P	probabilities corresponding to each of the potential pattern conditional on the other covariates denoted in x.
na.action	how to treat NAs
family	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See family for details of family functions.)
control	options for the optimization procedure (see optim, ucminf for options and details).
par	(optional) starting parameter vector
x	logical, add covariates matrix to result?
robust	logical, if true the computed asymptotic standard errors are replaced by their robust counterparts.

Examples

```
## simulate data

data <- simulate_GLM_dataset()


## estimate model without misclassification error

summary(lm(Y ~ X + M2, data))


## estimate model with misclassification error

summary(lm(Y ~ X + M, data))


## estimate misclassification probabilities

Pmodel <- glm(M2 ~ M + X, data = data, family = binomial("logit"))
```

```
summary(Pmodel)


## construct a-posteriori probabilities from Pmodel

P <- predict(Pmodel, newdata = data, type = "response")
P <- cbind(1 - P, P)
dimnames(P)[[2]] <- c("M0", "M1") ## speaking names


## estimate misclassGLM

est <- misclassGLM(Y = data$Y,
                   X = as.matrix(data[, 2, drop = FALSE]),
                   setM = matrix(c(0, 1), nrow = 2),
                   P = P)
summary(est)


## and bootstrapping the results from dataset
## Not run:
  summary(boot.misclassGLM(est,
                           Y = data$Y,
                           X = data.matrix(data[, 2, drop = FALSE]),
                           Pmodel = Pmodel,
                           PX = data,
                           repetitions = 100))

## End(Not run)

## simulate data

data <- simulate_GLM_dataset()


## estimate model without misclassification error

summary(lm(Y ~ X + M2, data))


## estimate model with misclassification error

summary(lm(Y ~ X + M, data))


## estimate misclassification probabilities

Pmodel <- glm(M2 ~ M + X, data = data, family = binomial("logit"))
summary(Pmodel)


## construct a-posteriori probabilities from Pmodel
```

```
P <- predict(Pmodel, newdata = data, type = "response")
P <- cbind(1 - P, P)
dimnames(P)[[2]] <- c("M0", "M1") ## speaking names


## estimate misclassGLM

est <- misclassGLM(Y = data$Y,
                   X = as.matrix(data[, 2, drop = FALSE]),
                   setM = matrix(c(0, 1), nrow = 2),
                   P = P)
summary(est)


## and bootstrapping the results from dataset
## Not run:
  summary(boot.misclassGLM(est,
                           Y = data$Y,
                           X = data.matrix(data[, 2, drop = FALSE]),
                           Pmodel = Pmodel,
                           PX = data,
                           repetitions = 100))

## End(Not run)
```

`misclassMlogit` *Mlogit estimation under misclassified covariate*

Description

`misclassMLogit` computes estimator for a GLM with a misclassified covariate using additional side information on the misclassification process

Usage

```
misclassMlogit(
  Y,
  X,
  setM,
  P,
  na.action = na.omit,
  control = list(),
  par = NULL,
  baseoutcome = NULL,
  x = FALSE
)
```

Arguments

Y	a matrix of 0s and 1s, indicating the target class. This is the dependent variable.
X	a matrix containing the independent variables
setM	matrix, rows containing potential patterns for a misclassed (latent) covariate M in any coding for a categorical independent variable, e.g. dummy coding.
P	probabilities corresponding to each of the potential pattern conditional on the other covariates denoted in x.
na.action	how to treat NAs
control	options for the optimization procedure (see optim, ucminf for options and details).
par	(optional) starting parameter vector
baseoutcome	reference outcome class
x	logical, add covariates matrix to result?

Examples

```
## simulate data

data <- simulate_mlogit_dataset()


## estimate model without misclassification error

library(mlogit)
data2 <- mlogit.data(data, varying = NULL, choice = "Y", shape = "wide")
summary(mlogit(Y ~ 1 | X + M2, data2, reflevel = "3"))


## estimate model with misclassification error

summary(mlogit(Y ~ 1 | X + M, data2, reflevel = "3"))


## estimate misclassification probabilities

Pmodel <- glm(M2 ~ M + X, data = data, family = binomial("logit"))
summary(Pmodel)


## construct a-posteriori probabilities from Pmodel

P <- predict(Pmodel, newdata = data, type = "response")
P <- cbind(1 - P, P)
dimnames(P)[[2]] <- c("M0", "M1") ## speaking names


## estimate misclassGLM
```

```
Yneu <- matrix(rep.int(0, nrow(data) * 3), ncol = 3)
for (i in 1:nrow(data)) Yneu[i, data$Y[i]] <- 1
est <- misclassMlogit(Y = Yneu,
                      X = as.matrix(data[, 2, drop = FALSE]),
                      setM = matrix(c(0, 1), nrow = 2),
                      P = P)
summary(est)


## and bootstrapping the results from dataset
## Not run:
summary(boot.misclassMlogit(est,
                         Y = Yneu,
                         X = data.matrix(data[, 2, drop = FALSE]),
                         Pmodel = Pmodel,
                         PX = data,
                         repetitions = 100))

## End(Not run)
```

`predict.misclassGLM` *Predict Method for* `misclassGLM` *Fits*

Description

Obtains predictions

Obtains predictions

Usage

```
## S3 method for class 'misclassGLM'
       ## S3 method for class 'misclassGLM'
predict(object, X, P = NULL, type = c("link", "response"),
                                     na.action = na.pass, ...)

## S3 method for class 'misclassGLM'
       ## S3 method for class 'misclassGLM'
predict(object, X, P = NULL, type = c("link", "response"),
                                     na.action = na.pass, ...)
```

Arguments

`object`	a fitted object of class inheriting from 'misclassGLM'.
`X`	matrix of fixed covariates
`P`	a-posteriori probabilities for the true values of the misclassified variable. If provided, the conditional expectation on X,P is computed, otherwise a set of marginal predictions is provided, one for each alternative.

type	the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities. The value of this argument can be abbreviated.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
...	additional arguments (not used at the moment)

See Also

misclassGLM

misclassGLM

`predict.misclassMlogit`

Predict Method for `misclassMlogit` *Fits*

Description

Obtains predictions

Usage

```
## S3 method for class 'misclassMlogit'
       ## S3 method for class 'misclassMlogit'
predict(object, X, P = NULL, type = c("link", "response"),
       na.action = na.pass, ...)
```

Arguments

object	a fitted object of class inheriting from 'misclassMlogit'.
X	matrix of fixed covariates.
P	a-posteriori probabilities for the true values of the misclassified variable. If provided, the conditional expectation on X,P is computed, otherwise a set of marginal predictions is provided, one for each alternative.
type	the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities. The value of this argument can be abbreviated.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
...	additional arguments (not used at the moment)

See Also

misclassMlogit

---

simulate_GLM_dataset *Simulate a Data Set to Use With* misclassGLM

Description

simulates a data set with - one continuous variable X drawn from a Gaussian distribution, - a binary or trinary variable M with misclassification (M2) - a dependent variable either with added Gaussian noise or drawn from a logit distribution

simulates a data set with - one continuous variable X drawn from a Gaussian distribution, - a binary or trinary variable M with misclassification (M2) - a dependent variable either with added Gaussian noise or drawn from a logit distribution

Usage

```
simulate_GLM_dataset(
  n = 50000,
  const = 0,
  alpha = 1,
  beta = -2,
  beta2 = NULL,
  logit = FALSE
)

simulate_GLM_dataset(
  n = 50000,
  const = 0,
  alpha = 1,
  beta = -2,
  beta2 = NULL,
  logit = FALSE
)
```

Arguments

n	number observations
const	constant
alpha	parameter for X
beta	parameter for M(1)
beta2	parameter for M2, if NULL, M is a binary covariate, otherwise a three-valued categorical
logit	logical, if true logit regression, otherwise Gaussian regression

Details

This can be used to demonstrate the abilities of misclassGLM. For an example see misclassGLM.

This can be used to demonstrate the abilities of misclassGLM. For an example see misclassGLM.

See Also

misclassGLM

misclassGLM

---

simulate_mlogit_dataset

Simulate a Data Set to Use With misclassMlogit

Description

simulates a data set with - one continuous variable X drawn from a Gaussian distribution, - a binary or trinary variable M with misclassification (M2) - a dependent variable drawn from a multionomial distribution dependent on X and M.

Usage

```
simulate_mlogit_dataset(
  n = 1000,
  const = c(0, 0),
  alpha = c(1, 2),
  beta = -2 * c(1, 2),
  beta2 = NULL
)
```

Arguments

n	number observations
const	constants
alpha	parameters for X
beta	parameters for M(1)
beta2	parameters for M2, if NULL, M is a binary covariate, otherwise a three-valued categorical.

Details

This can be used to demonstrate the abilities of misclassMlogit. For an example see misclassMlogit.

See Also

misclassMlogit

Index