Package 'flipscores'

September 12, 2018

Type Package

Title Robust Testing in GLMs

Version 0.2

Date 2018-09-04

Author Jesse Hemerik, Jelle Goeman and Livio Finos

Maintainer Jesse Hemerik <j.b.a.hemerik@lumc.nl>

Description Provides robust tests for testing in GLMs,
by sign-flipping score contributions.
The tests are often robust against overdispersion,
heteroscedasticity and, in some cases,
ignored nuisance variables.
See Hemerik and Goeman (2017) <doi:10.1007/s11749-017-0571-1>.

License GNU General Public License

LazyData TRUE

Imports stats, methods

NeedsCompilation no

Repository CRAN

Date/Publication 2018-09-12 15:40:03 UTC

R topics documented:

flipscores 2
flipscoreshd 3

Index **5**

`flipscores` *Robust testing in GLMs*

Description

Provides two robust tests for testing in GLMs, by sign-flipping score contributions. The tests are often robust against overdispersion, heteroscedasticity and, in some cases, ignored nuisance variables.

Usage

```
flipscores(model0, model1, X1, alternative = "two.sided",  w=1E5, scoretype="basic")
```

Arguments

`model0`	A glm or lm object representing the fit under H0.
`model1`	A glm or lm object representing the fit under H1. It contains one covariate in addition to those in model0. If that covariate is a factor, it should have at most 2 levels. Either `model1` or `X1` should be specified.
`X1`	Vector with the values of the covariate of interest. If X1 is a factor, it should have at most 2 levels. Either `model1` or `X1` should be specified.
`alternative`	Should be `"greater"`, `"less"` or `"two.sided"`
`w`	The number of times that the scores are randomly sign-flipped.
`scoretype`	The type of score that is computed, either `"basic"` or `"effective"`. Using `"effective"` takes into account nuisance estimation.

Value

A p-value.

Examples

```
set.seed(8153)
n <-50
beta <- 2  #coeffient of interest (H0: beta=0)
gamma <- 1   #nuisance coefficient

dataset <- data.frame(x=NULL,z=NULL,y=NULL)
dataset[n,] <- NA
dataset[,1:2] <- data.frame(matrix(rnorm(n*2),n,2))  #generate covariates
colnames(dataset) <- c("x","z")
dataset$y <- rnbinom(n, mu = exp(dataset[,1]*beta + dataset[,2]*gamma), size=1 )

#Y has a negative binomial distribution but we assume a Poisson model:
modelz <- glm(y~z, family=poisson,data=dataset,x=TRUE)
```

```
## Basic test:
pv1 <- flipscores(model0=modelz, X1=dataset[,1], alternative = "two.sided", scoretype="basic")
pv1  #p-value

## Test that takes into account nuisance estimation:
pv2 <- flipscores(model0=modelz, X1=dataset[,1], alternative = "two.sided", scoretype="eff")
pv2  #p-value
```

`flipscoreshd` *Robust testing in GLMs*

Description

Similar to the flipscores() function, but allows a multi-dimensional (even high-dimensional) parameter of interest.

Usage

```
flipscoreshd(model0, X1,  w=200, scoretype="basic", V="identity")
```

Arguments

`model0`	A glm or lm object representing the fit under H0.
`X1`	n by d dataframe or matrix with the d tested covariates (which have no effect under H0).
`w`	The number of times that the scores are randomly sign-flipped.
`scoretype`	The type of score that is used, either `"basic"` or `"effective"`. Using `"effective"` takes into account nuisance estimation.
`V`	User-defined d-by-d matrix, where d is the dimension of the parameter of interest, i.e. the nr of columns of X1. One can also specify `V="identity"` to use the identity matrix, or specify `V="invinfo"` to use the inverse of the effective Fisher info.

Value

A p-value.

Examples

```
set.seed(8193)
n <-50

dataset= data.frame(matrix(rnorm(n*5),n,5))

for(k in 1:n){
```

```
  dataset[k,1:5] <- dataset[k,1:5] + rnorm(1)   #make the covariates correlated
}

#Y has a negative binomial distribution but we assume a Poisson model:
#Y depends only on the 4-th covariate:
dataset$y <- rnbinom(n, mu = exp(dataset[,4]*0.5), size=1 )

#We will test the hypothesis H0 that X1,X2,X3 all have coefficient 0.
#So H0 is true. X4,X5 are nuisance covariates.

formula0 <- as.formula("y ~ X4+X5")           #X4,X5: nuisance
X1 <- dataset[,1:3]                           #X1,X2,X3: tested

model0 = glm(formula0, family=poisson,na.action=na.omit,data=dataset,x=TRUE)

## Basic test (often conservative):
pv1 <- flipscoreshd(model0=model0, X1=X1, w=200, scoretype="basic",V="invinfo")
pv1  #p-value

## Test that takes into account nuisance estimation:
pv1 <- flipscoreshd(model0=model0, X1=X1, w=200, scoretype="effective",V="invinfo")
pv1  #p-value
```

Index

flipscores, 2
flipscoreshd, 3