Package 'GLMaSPU'

October 12, 2022

Type Package

Title An Adaptive Test on High Dimensional Parameters in Generalized
Linear Models

Version 1.0

Date 2016-10-10

Author Chong Wu and Wei Pan

Maintainer Chong Wu <wuxx0845@umn.edu>

Description Several tests for high dimensional generalized linear models have been proposed recently. In this package, we implemented a new test called adaptive sum of powered score (aSPU) for high dimensional generalized linear models, which is often more powerful than the existing methods in a wide scenarios. We also implemented permutation based version of several existing methods for research purpose. We recommend users use the aSPU test for their real testing problem. You can learn more about the tests implemented in the package via the following papers: 1. Pan, W., Kim, J., Zhang, Y., Shen, X. and Wei, P. (2014) <DOI:10.1534/genetics.114.165035> A powerful and adaptive association test for rare variants, Genetics, 197(4). 2. Guo, B., and Chen, S. X. (2016) <DOI:10.1111/rssb.12152>. Tests for high dimensional generalized linear models. Journal of the Royal Statistical Society: Series B. 3. Goeman, J. J., Van Houwelingen, H. C., and Finos, L. (2011) <DOI:10.1093/biomet/asr016>. Testing against a high-dimensional alternative in the generalized linear model: asymptotic type I error control. Biometrika, 98(2).

License GPL-2

Imports MASS, mnormt, Rcpp (>= 0.12.6)

Depends R (>= 3.1.1), mvtnorm (>= 1.0-0)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-12-09 08:58:42

R topics documented:

GLMaSPU-package *An Adaptive Test on High Dimensional Parameters in Generalized Linear Models*

Description

Provide various tests on high-dimensional parameters in generalized linear models.

Details

Several tests for high dimensional generalized linear models have been proposed recently. In this package, we implemented a new test called aSPU for high dimensional generalized linear models, which is often more powerful than the existing methods in a wide scenarios. We also implemented permutation based version of several existing methods for research purpose. We recommend users use the aSPU test for their real testing problem.

Author(s)

Chong Wu, Wei Pan Maintainer: Chong Wu <wuxx0845@umn.edu>

References

Chong Wu, Gongjun Xu and Wei Pan, "An Adaptive test on high dimensional parameters in generalized linear models" (Submitted)

aSPU_apval *Asymptotic based Sum of Powered Score (SPU) tests and adaptive SPU (aSPU) test.*

Description

It gives p-values of the SPU tests and aSPU test.

Usage

```
aSPU_apval(Y, X, cov = NULL, pow = c(1:6, Inf), resample = "boot",
  model = "gaussian", n.perm = 5000)
```

Arguments

`Y` Response. It can be binary or continuous trait. A vector with length n (number of observations).

`X` Genotype or other data; each row for a subject, and each column for a variable of interest. An n by p matrix (n: number of observations, p: number of predictors).

`cov` Covariates. An n by q matrix (n: number of observations, q: number of covariates).

`pow` Gamma set used in SPU test. A vector of the powers.

`resample` Resample methods. "perm" for residual permutations; "boot" for parametric bootstrap.

`model` corresponding to the Response. "gaussian" for a quantitative response; "binomial" for a binary response.

`n.perm` number of permutations or bootstraps.

Value

A list object, Ts : test statistics for the SPU tests and the aSPU test. pvs : p-values for the SPU and aSPU tests.

Author(s)

Chong Wu and Wei Pan

References

Chong Wu, Gongjun Xu and Wei Pan, "An Adaptive test on high dimensional parameters in generalized linear models" (Submitted)

Examples

```
p = 200
n = 100
beta = c(1,3,3)
s = 0.15
non.zero = floor(p * s)
signal.r = 0.08
seed = 2
alpha = c(rep(signal.r,non.zero),rep(0,p-non.zero))
dat = generate_data(seed, n = n, p = p, beta = beta,alpha = alpha)
cov = dat$Z
X = dat$X
Y = dat$Y
aSPU_apval(Y, X, cov = cov, pow = c(1:6, Inf),resample = "perm", model = "gaussian",  n.perm = 1000)
# The p-values are similar to the resample based one
```

`aSPU_perm` *Resample based Sum of Powered Score (SPU) tests and adaptive SPU (aSPU) test.*

Description

`aSPU_perm` returns p-values of the SPU tests and aSPU test.

Usage

```
aSPU_perm(Y, X, cov = NULL, resample = c("perm", "boot"),
  model = c("gaussian", "binomial"), pow = c(1:6, Inf), n.perm = 1000)
```

Arguments

`Y` Response. It can be binary or continuous trait. A vector with length n (number of observations).

`X` Genotype or other data; each row for a subject, and each column for a variable of interest. An n by p matrix (n: number of observations, p: number of predictors).

`cov` Covariates. An n by q matrix (n: number of observations, q: number of covariates).

`resample` Resample methods. "perm" for residual permutations; "boot" for parametric bootstrap.

`model` corresponding to the Response. "gaussian" for a quantitative response; "binomial" for a binary response.

`pow` Gamma set used in SPU test. A vector of the powers.

`n.perm` number of permutations or bootstraps.

Value

A list object, Ts : test statistics for the SPU tests and the aSPU test. pvs : p-values for the SPU and aSPU tests.

Author(s)

Chong Wu and Wei Pan

References

Wei Pan, Junghi Kim, Yiwei Zhang, Xiaotong Shen and Peng Wei (2014) A powerful and adaptive association test for rare variants, Genetics, 197(4), 1081-95

Examples

```
p = 200
n = 100
beta = c(1,3,3)
s = 0.15
signal.r = 0.08
non.zero = floor(p * s)
seed = 2
alpha = c(rep(signal.r,non.zero),rep(0,p-non.zero))
dat = generate_data(seed, n = n, p = p, beta = beta,alpha = alpha)
cov = dat$Z
X = dat$X
Y = dat$Y
aSPU_perm(Y, X, cov = cov, pow = c(1:6, Inf),resample = "perm", model = "gaussian",  n.perm = 1000)
# The p-values are similar to the asymptotic based one
```

`generate_data` *Generate data for generalized linear models in simulation.*

Description

`generate_data` returns simulated data, including response Y, covariates Z, and variable of interest X.

Usage

```
generate_data(seed, n, p, beta, alpha)
```

Arguments

`seed`	Random seed.
`n`	Number of samples
`p`	Dimension of variable of interest
`beta`	Coefficients for covariates Z
`alpha`	Coefficients for variable of interest X

Value

A list object

Author(s)

Chong Wu and Wei Pan

References

Chong Wu, Gongjun Xu and Wei Pan, "An Adaptive test on high dimensional parameters in generalized linear models" (Submitted)

Examples

```
p = 200
n = 100
beta = c(1,3,3)
s = 0.15
signal.r = 0.08
non.zero = floor(p * s)
seed = 2
alpha = c(rep(signal.r,non.zero),rep(0,p-non.zero))
dat = generate_data(seed, n = n, p = p, beta = beta,alpha = alpha)
#X, Y, cov
#dat$X; dat$Y; dat$cov
```

Goeman_perm *Resample based Goeman test.*

Description

`Goeman_perm` returns resample based p-value for a test proposed by Goeman (2011).

Usage

```
Goeman_perm(Y, X, cov = NULL, model = c("gaussian", "binomial"),
  n.perm = 1000)
```

Arguments

`Y`	Response. It can be binary or continuous trait. A vector with length n (number of observations).
`X`	Genotype or other data; each row for a subject, and each column for a variable of interest. An n by p matrix (n: number of observations, p: number of predictors).
`cov`	Covariates. An n by q matrix (n: number of observations, q: number of covariates).
`model`	corresponding to the Response. "gaussian" for a quantitative response; "binomial" for a binary response.
`n.perm`	number of permutations or bootstraps.

Details

Goeman_perm calculates the resample based p-value. You can calculate the asymptotic based p-value via using R package globaltest. Based on our experience, resample based p-value is often different from the asymptotic based one, except when the dimension of X is larger than the sample size n.

Value

A list object, Ts : test statistics for the SPU tests and the aSPU test. pvs : p-values for the SPU and aSPU tests.

Author(s)

Chong Wu and Wei Pan

References

Goeman, J. J., Van Houwelingen, H. C. and Finos, L. (2011). Testing against a high-dimensional alternative in the generalized linear model asymptotic type 1 error control. Biometrika, 98(2), 381-390.

Examples

```
p = 200
n = 100
beta = c(1,3,3)
s = 0.15
signal.r = 0.08
non.zero = floor(p * s)
seed = 2
alpha = c(rep(signal.r,non.zero),rep(0,p-non.zero))
dat = generate_data(seed, n = n, p = p, beta = beta,alpha = alpha)
cov = dat$Z
X = dat$X
Y = dat$Y
Goeman_perm(Y, X, cov = cov,model="gaussian", n.perm=1000)
```

HDGLM_perm *Resample based HDGLM test.*

Description

HDGLM_perm returns resample based p-value for HDGLM test (Guo 2016).

Usage

```
HDGLM_perm(Y, X, cov = NULL, model = c("gaussian", "binomial"),
  n.perm = 1000)
```

Arguments

Y Response. It can be binary or continuous trait. A vector with length n (number of observations).

X Genotype or other data; each row for a subject, and each column for a variable of interest. An n by p matrix (n: number of observations, p: number of predictors).

cov Covariates. An n by q matrix (n: number of observations, q: number of covariates).

model corresponding to the Response. "gaussian" for a quantitative response; "binomial" for a binary response.

n.perm number of permutations or bootstraps.

Details

HDGLM_perm calculates the resample based p-value. You can calculate the asymptotic based p-value by using HDGLM_test function in R package HDGLM. Based on our experience, resample based p-value is often similar to the asymptotic based one, except when the signals are highly sparse.

Value

A list object, Ts : test statistics for the SPU tests and the aSPU test. pvs : p-values for the SPU and aSPU tests.

Author(s)

Chong Wu and Wei Pan

References

Guo, B. and S. X. Chen (2016). Tests for high dimensional generalized linear models. Journal of the Royal Statistical Society: Series B (Statistical Methodology).

Examples

```
p = 200
n = 100
beta = c(1,3,3)
s = 0.15
signal.r = 0.08
seed = 2
non.zero = floor(p * s)
alpha = c(rep(signal.r,non.zero),rep(0,p-non.zero))
dat = generate_data(seed, n = n, p = p, beta = beta,alpha = alpha)
cov = dat$Z
```

```
X = dat$X
Y = dat$Y
HDGLM_perm(Y, X, cov = cov, model = "gaussian",  n.perm = 1000)
```

Index