Package 'SIHR'

October 7, 2021

Type Package

Title Statistical Inference in High Dimensional Regression

Version 0.2.0

Description Inference procedures in the high-dimensional setting for (1) linear functionals and quadratic functionals in linear regression ('Cai et al.' (2019) <arXiv:1904.12891>, 'Guo et al.' (2019) <arXiv:1909.01503>), (2) linear functional in logistic regression ('Guo et al.' <arXiv:2012.07133>), (3) individual treatment effects in linear and logistic regression, (4) single regression coefficient in binary outcome regression.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

Imports CVXR, Matrix, glmnet, Rdpack, scalreg, qpdf

RdMacros Rdpack

Suggests knitr, rmarkdown, MASS

URL `https://github.com/prabrishar1/SIHR`

BugReports `https://github.com/prabrishar1/SIHR/issues`

VignetteBuilder knitr

NeedsCompilation no

Author Prabrisha Rakshit [aut, cre],
Tony Cai [aut],
Zijian Guo [aut]

Maintainer Prabrisha Rakshit <prabrisha.rakshit@rutgers.edu>

Repository CRAN

Date/Publication 2021-10-07 08:00:08 UTC

R topics documented:

`Direction_fixedtuning` *Constructs the projection direction with a fixed tuning parameter*

Description

Constructs the projection direction, used for bias-correction, with a fixed tuning parameter

Usage

```
Direction_fixedtuning(
  X,
  loading,
  mu = NULL,
  model = "linear",
  weight = NULL,
  deriv.vec = NULL
)
```

Arguments

`X`	Design matrix, of dimension n x p
`loading`	Loading, of length p
`mu`	The dual tuning parameter used in the construction of the projection direction
`model`	The high dimensional regression model, either `linear` or `logistic` (default = `linear`)
`weight`	The weight vector of length n; to be supplied if `model="logistic"` (default=`NULL` when `model=linear`)
`deriv.vec`	The first derivative vector of the logit function at $X\%*\%$(`init.coef`), of length n ; to be supplied if `model="logistic"`. Here `init.coef` is the initial estimate of the regression vector. (default = `NULL` when `model=linear`)

Value

`proj`	The projection direction, of length p

Examples

```
n <- 100
p <- 400
X <- matrix(sample(-2:2,n*p,replace = TRUE),nrow = n,ncol = p)
resol <- 1.5
step <- 3
Est <- Direction_fixedtuning(X,loading=c(1,rep(0,(p-1))),mu=sqrt(2.01*log(p)/n)*resol^{-(step-1)})
```

Direction_searchtuning

Searches for the best step size and computes the projection direction with the searched best step size

---

Description

Searches for the best step size and computes the projection direction with the searched best step size

Usage

```
Direction_searchtuning(
  X,
  loading,
  model = "linear",
  weight = NULL,
  deriv.vec = NULL,
  resol = 1.5,
  maxiter = 6
)
```

Arguments

`X`	Design matrix, of dimension n x p
`loading`	Loading, of length p
`model`	The high dimensional regression model, either `linear` or `logistic` (default = `linear`)
`weight`	The weight vector of length n; to be supplied if `model="logistic"` (default=`NULL` when `model=linear`)
`deriv.vec`	The first derivative vector of the logit function at $X\%*\%$(`init.coef`), of length n ; to be supplied if `model="logistic"`. Here `init.coef` is the initial estimate of the regression vector. (default = `NULL` when `model=linear`)
`resol`	The factor by which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 1.5)
`maxiter`	Maximum number of steps along which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 6)

Value

`proj`	The projection direction, of length p
`step`	The best step size

Examples

```
n <- 100
p <- 400
X <- matrix(sample(-2:2,n*p,replace = TRUE),nrow = n,ncol = p)
Est <- Direction_searchtuning(X,loading=c(1,rep(0,(p-1))))
```

`GLM_binary` *Inference for single regression coefficient in high dimensional probit regression model*

Description

Computes the bias corrected estimator of a single regression coefficient in the high dimensional binary outcome regression model and the corresponding standard error. It also constructs the confidence interval for the target regression coefficient and tests whether it is equal to a pre-specified value `b0`.

Usage

```
GLM_binary(
  X,
  y,
  index,
  model = "logistic1",
  intercept = TRUE,
  init.coef = NULL,
  lambda = NULL,
  mu = NULL,
  step = NULL,
  resol = 1.5,
  maxiter = 6,
  b0 = 0,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`X`	Design matrix, of dimension n x p
`y`	Outcome vector, of length n

index	An integer between 1 and p indicating the index of the targeted regression coefficient. For example, index $= 1$ means that the first regression coefficient is our inference target
model	The fitted GLM, either logistic1 or logistic2 or probit or inverse t1 (default = logistic1) ; model="logistic1" uses SIHR::LF_logistic with weight=NULL; model="logistic2" uses SIHR::LF_logistic with weight=rep(1,n)
intercept	Should intercept be fitted for the initial estimator (default = TRUE)
init.coef	Initial estimator of the regression vector (default = NULL)
lambda	The tuning parameter used in the construction of init.coef (default = NULL)
mu	The dual tuning parameter used in the construction of the projection direction (default = NULL)
step	The step size used to compute mu; if set to NULL it is computed to be the number of steps (< maxiter) to obtain the smallest mu such that the dual optimization problem for constructing the projection direction converges (default = NULL)
resol	The factor by which mu is increased/decreased to obtain the smallest mu such that the dual optimization problem for constructing the projection direction converges (default = 1.5)
maxiter	Maximum number of steps along which mu is increased/decreased to obtain the smallest mu such that the dual optimization problem for constructing the projection direction converges (default = 6)
b0	The null value to be tested against
alpha	Level of significance to test the null hypothesis that the target regression coefficient is equal to b0 (default = 0.05)
verbose	Should inetrmediate message(s) be printed (default = TRUE)

Value

prop.est	The bias corrected estimator of the target regression coefficient
se	The standard error of the bias-corrected estimator
CI	The confidence interval for the target regression coefficient
decision	decision$= 1$ implies the target regression coefficient is not equal to b0 decision$= 0$ implies the target regression coefficient is equal to b0
proj	The projection direction, of length p

References

Cai TT, Guo Z, Ma R (To Appear). "Statistical Inference for High-Dimensional Generalized Linear Models with Binary Outcomes." *Journal of the American Statistical Association*.

Examples

```
sp = 20
n = 400
p = 800
```

```
f = function(x){
pnorm(x)
}
sig1 = toeplitz(seq(0.6, 0,length.out = p/10))
Sig = Matrix::bdiag(rep(list(sig1),10))+diag(rep(0.4,p))
X = MASS::mvrnorm(n, mu=rep(0,p), Sigma=Sig)
b = rep(0,p)
b[1:sp] = rep(c(0.4,-0.4), sp/2)
prob = f(X %*% b)
y = array(dim = 1)
for(i in 1:n){
y[i] = rbinom(1,1,prob[i])
}
Est = SIHR::GLM_binary(X = X, y = y, index = 1, model = "probit", intercept = FALSE)
```

ITE *Individualized treatment effect in the high dimensional linear regression*

Description

Computes the bias corrected estimator of the Individualized Treatment Effect (ITE) and the corresponding standard error. It also constructs the confidence interval for ITE and tests whether ITE is above zero or not. Here ITE is defined as a linear combination of the difference between two regression vectors.

Usage

```
ITE(
  X1,
  y1,
  X2,
  y2,
  loading,
  intercept.loading = TRUE,
  intercept = TRUE,
  init.coef1 = NULL,
  init.coef2 = NULL,
  lambda1 = NULL,
  lambda2 = NULL,
  mu1 = NULL,
  mu2 = NULL,
  step1 = NULL,
  step2 = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

X1	Design matrix for the first sample, of dimension n_1 x p
y1	Outcome vector for the first sample, of length n_1
X2	Design matrix for the second sample, of dimension n_2 x p
y2	Outcome vector for the second sample, of length n_2
loading	Loading, of length p
intercept.loading	Should intercept be included for the loading (default = TRUE)
intercept	Should intercept(s) be fitted for the initial estimators(default = TRUE)
init.coef1	Initial estimator of the first regression vector (default = NULL)
init.coef2	Initial estimator of the second regression vector (default = NULL)
lambda1	The tuning parameter in the construction of init.coef1 (default = NULL)
lambda2	The tuning parameter in the construction of init.coef2 (default = NULL)
mu1	The dual tuning parameter used in the construction of the first projection direction (default = NULL)
mu2	The dual tuning parameter used in the construction of the second projection direction (default = NULL)
step1	The step size used to compute mu1; if set to NULL it is computed to be the number of steps (< maxiter) to obtain the smallest mu1 such that the dual optimization problem for constructing the first projection direction converges (default = NULL)
step2	The step size used to compute mu2; if set to NULL it is computed to be the number of steps (< maxiter) to obtain the smallest mu2 such that the dual optimization problem for constructing the second projection direction converges (default = NULL)
resol	The factor by which mu1 (and mu2) is increased/decreased to obtain the smallest mu1 (and mu2) such that the dual optimization problem for constructing the first (and the second) projection direction converges (default = 1.5)
maxiter	Maximum number of steps along which mu1 (and mu2) is increased/decreased to obtain the smallest mu1 (and mu2) such that the dual optimization problem for constructing the first (and the second) projection direction converges (default = 6)
alpha	Level of significance to test the null hypothesis which claims that ITE is not above zero (default = 0.05)
verbose	Should inetrmediate message(s) be printed (default = TRUE)

Value

prop.est	The bias-corrected estimator of the ITE
se	The standard error of the bias-corrected estimator
CI	The confidence interval for the ITE
decision	decision$=1$ implies the ITE is above zero decision$=0$ implies the ITE is not above zero

References

Cai T, Cai TT, Guo Z (2019). "Optimal Statistical Inference for Individualized Treatment Effects in High-dimensional Models." *Journal of the Royal Statistical Society: Series B*. https://arxiv.org/pdf/1904.12891.pdf.

Examples

```
n1 <- 90
p <- 200
n2 <- 90
mu <- rep(0,p)
beta1 <- rep(0,p)
beta1[1:10] <- c(1:10)/5
beta2 <- rep(0,p)
beta2[1:5] <- c(1:5)/10
X1 <- MASS::mvrnorm(n1, mu, diag(p))
X2 <- MASS::mvrnorm(n2, mu, diag(p))
y1 <- X1%*%beta1 + rnorm(n1)
y2 <- X2%*%beta2 + rnorm(n2)
loading <- c(1,rep(0, (p-1)))
Est <- ITE(X1 = X1, y1 = y1, X2 = X2, y2 = y2,loading = loading)
```

`ITE_Logistic` *Inference for difference of case probabilities in high dimensional logistic regressions*

Description

Computes the bias corrected estimator of the difference between case probabilities or a linear combination of the difference between two regression vectors with respect to two high dimensional logistic regression models and the corresponding standard error. It also constructs the confidence interval for the difference of case probabilities or a linear combination of the difference between the regression vectors and test whether it is above zero or not. Here the case probability refers to the conditional probability of the binary response variable taking value 1 given the predictors are assigned to `loading`.

Usage

```
ITE_Logistic(
  X1,
  y1,
  X2,
  y2,
  loading,
  weight = NULL,
  trans = TRUE,
```

```
  intercept = TRUE,
  intercept.loading = TRUE,
  init.coef1 = NULL,
  init.coef2 = NULL,
  lambda1 = NULL,
  lambda2 = NULL,
  mu1 = NULL,
  mu2 = NULL,
  step1 = NULL,
  step2 = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

- `X1` Design matrix for the first sample, of dimension n_1 x p
- `y1` Outcome vector for the first sample, of length n_1
- `X2` Design matrix for the second sample, of dimension n_2 x p
- `y2` Outcome vector for the second sample, of length n_2
- `loading` Loading, of length p
- `weight` The weight vector used for bias correction, of length n; if set to `NULL`, the weight is the inverse of the first derivative of the logit function (default = `NULL`)
- `trans` Should results for the case probability (`TRUE`) or the linear combination (`FALSE`) be reported (default = `TRUE`)
- `intercept` Should intercept(s) be fitted for the initial estimators (default = `TRUE`)
- `intercept.loading` Should intercept be included for the `loading` (default = `TRUE`)
- `init.coef1` Initial estimator of the first regression vector (default = `NULL`)
- `init.coef2` Initial estimator of the second regression vector (default = `NULL`)
- `lambda1` The tuning parameter in the construction of `init.coef1` (default = `NULL`)
- `lambda2` The tuning parameter in the construction of `init.coef2` (default = `NULL`)
- `mu1` The dual tuning parameter used in the construction of the first projection direction (default = `NULL`)
- `mu2` The dual tuning parameter used in the construction of the second projection direction (default = `NULL`)
- `step1` The step size used to compute `mu1`; if set to `NULL` it is computed to be the number of steps (< `maxiter`) to obtain the smallest `mu1` such that the dual optimization problem for constructing the projection direction converges (default = `NULL`)
- `step2` The step size used to compute `mu2`; if set to `NULL` it is computed to be the number of steps (< `maxiter`) to obtain the smallest `mu2` such that the dual optimization problem for constructing the second projection direction converges (default = `NULL`)

resol	The factor by which mu1 (and mu2) is increased/decreased to obtain the smallest mu1 (and mu2) such that the dual optimization problem for constructing the first (and the second) projection direction converges (default = 1.5)
maxiter	Maximum number of steps along which mu1 (and mu2) is increased/decreased to obtain the smallest mu (and mu2) such that the dual optimization problem for constructing the first (and the second) projection direction converges (default = 6)
alpha	Level ofsignificance to test the null hypothesis which claims that the first case probability is not greater than the second case probability (default = 0.05)
verbose	Should inetrmediate message(s) be printed (default = TRUE)

Value

prop.est	The bias-corrected estimator for the difference between case probabilities or the linear combination of the difference between two regression vectors
se	The standard error for the bias-corrected estimator
CI	The confidence interval for the difference between case probabilities or the linear combination of the difference between two regression vectors
decision	decision= 1 implies the first case probability or linear combination is greater than the second one decision= 0 implies the first case probability or linear combination is less than the second one

Examples

```
A1gen <- function(rho,p){
A1 <- matrix(0,p,p)
for(i in 1:p){
  for(j in 1:p){
    A1[i,j] <- rho^(abs(i-j))
  }
}
A1
}
n1 <- 100
n2 <- 100
p <- 400
mu <- rep(0,p)
rho <- 0.5
Cov <- (A1gen(rho,p))/2
beta1 <- rep(0,p)
beta1[1:10] <- c(1:10)/5
beta2 <- rep(0,p)
beta2[1:5] <- c(1:5)/10
X1 <- MASS::mvrnorm(n1,mu,Cov)
X2 <- MASS::mvrnorm(n2,mu,Cov)
exp_val1 <- X1%*%beta1
exp_val2 <- X2%*%beta2
```

```
prob1 <- exp(exp_val1)/(1+exp(exp_val1))
prob2 <- exp(exp_val2)/(1+exp(exp_val2))
y1 <- rbinom(n1,1,prob1)
y2 <- rbinom(n2,1,prob2)
loading <- c(1,rep(0,(p-1)))
Est <- ITE_Logistic(X1 = X1, y1 = y1, X2 = X2, y2 = y2,loading = loading, trans = FALSE)
```

LF *Inference for a linear combination of regression coefficients in high dimensional linear regression.*

Description

Computes the bias corrected estimator of the linear combination of regression coefficients and the corresponding standard error. It also constructs the confidence interval for the linear combination and tests whether it is above zero or not.

Usage

```
LF(
  X,
  y,
  loading,
  intercept.loading = TRUE,
  intercept = TRUE,
  init.coef = NULL,
  lambda = NULL,
  mu = NULL,
  step = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`X`	Design matrix, of dimension n x p
`y`	Outcome vector, of length n
`loading`	Loading, of length p
`intercept.loading`	Should intercept be included for the `loading` (default = `TRUE`)
`intercept`	Should intercept be fitted for the initial estimator (default = `TRUE`)
`init.coef`	Initial estimator of the regression vector (default = `NULL`)
`lambda`	The tuning parameter in the construction of `init.coef` (default = `NULL`)

`mu`	The dual tuning parameter used in the construction of the projection direction (default = `NULL`)
`step`	The step size used to compute `mu`; if set to `NULL` it is computed to be the number of steps (< `maxiter`) to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = `NULL`)
`resol`	The factor by which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 1.5)
`maxiter`	Maximum number of steps along which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 6)
`alpha`	Level of significance to test the null hypothesis which claims that the linear combination of the regression coefficients is less than or equal to zero (default = 0.05)
`verbose`	Should inetrmediate message(s) be printed (default = `TRUE`)

Value

`prop.est`	The bias-corrected estimator for the linear combination of regression coefficients
`se`	The standard error of the bias-corrected estimator
`CI`	The confidence interval for the linear combination
`decision`	`decision`$=1$ implies the linear combination is above zero `decision`$=0$ implies the linear combination is not above zero
`proj`	The projection direction, of length p
`plug.in`	The plug-in estimator for the linear combination

References

Cai T, Cai TT, Guo Z (2019). "Optimal Statistical Inference for Individualized Treatment Effects in High-dimensional Models." *Journal of the Royal Statistical Society: Series B*. https://arxiv.org/pdf/1904.12891.pdf.

Examples

```
n <- 90
p <- 200
A1gen <- function(rho,p){
A1=matrix(0,p,p)
for(i in 1:p){
 for(j in 1:p){
   A1[i,j] <- rho^(abs(i-j))
 }
}
A1
}
mu <- rep(0,p)
```

```
rho <- 0.5
Cov <- (A1gen(rho,p))/2
beta <- rep(0,p)
beta[1:10] <- c(1:10)/5
X <- MASS::mvrnorm(n,mu,Cov)
y <- X%*%beta + rnorm(n)
loading <- c(1,rep(0,(p-1)))
Est <- LF(X = X, y = y, loading = loading)
```

`LF_logistic` *Inference for the case probability or a linear combination of regression coefficients in high dimensional logistic regression.*

Description

Computes the bias corrected estimator of the case probability or the linear combination of regression coefficients in the high dimensional logistic regression model and the corresponding standard error. It also constructs the confidence interval for the case probability or the linear combination and tests whether the case probability is above 0.5 or not. Here case probability refers to the conditional probability of the binary response variable taking value 1 given the predictors take value `loading`.

Usage

```
LF_logistic(
  X,
  y,
  loading,
  weight = NULL,
  trans = TRUE,
  intercept.loading = TRUE,
  intercept = TRUE,
  init.coef = NULL,
  lambda = NULL,
  mu = NULL,
  step = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

X	Design matrix, of dimension n x p
y	Outcome vector, of length n
loading	Loading, of length p

weight — The weight vector used for bias correction, of length n; if set to NULL, the weight is the inverse of the first derivative of the logit function (default = NULL)

trans — Should results for the case probability (TRUE) or the linear combination (FALSE) be reported (default = TRUE)

intercept.loading — Should intercept be included for the loading (default = TRUE)

intercept — Should intercept be fitted for the initial estimator (default = TRUE)

init.coef — Initial estimator of the regression vector (default = NULL)

lambda — The tuning parameter used in the construction of init.coef (default = NULL)

mu — The dual tuning parameter used in the construction of the projection direction (default = NULL)

step — The step size used to compute mu; if set to NULL it is computed to be the number of steps (< maxiter) to obtain the smallest mu such that the dual optimization problem for constructing the projection direction converges (default = NULL)

resol — The factor by which mu is increased/decreased to obtain the smallest mu such that the dual optimization problem for constructing the projection direction converges (default = 1.5)

maxiter — Maximum number of steps along which mu is increased/decreased to obtain the smallest mu such that the dual optimization problem for constructing the projection direction converges (default = 6)

alpha — Level of significance to test the null hypothesis that the case probability is less than or equal to 0.5 (default = 0.05)

verbose — Should inetrmediate message(s) be printed (default = TRUE)

Value

prop.est — The bias corrected estimator of the case probability or the linear combination of regression coefficients

se — The standard error of the bias-corrected estimator

CI — The confidence interval for the case probability or the linear combination

decision — decision$= 1$ implies the case probability is above 0.5
decision$= 0$ implies the case probability is not above 0.5

proj — The projection direction, of length p

plug.in — The plug-in estimator of the case probability or the linear combination

References

Guo Z, Rakshit P, Herman DS, Chen J (2019). "Inference for Case Probability in High-dimensional Logistic Regression." *Unknown*. https://arxiv.org/abs/2012.07133.

Examples

```
A1gen <- function(rho,p){
 A1 <- matrix(0,p,p)
 for(i in 1:p){
   for(j in 1:p){
     A1[i,j] <- rho^(abs(i-j))
   }
 }
 A1
}
n <- 100
p <- 400
mu <- rep(0,p)
rho <- 0.5
Cov <- (A1gen(rho,p))/2
beta <- rep(0,p)
beta[1:10] <- 0.5*c(1:10)/10
a0 <- 0
loading <- c(1,rep(0,(p-1)))
X <- MASS::mvrnorm(n,mu,Cov)
exp_val <- X%*%beta+a0
prob <- exp(exp_val)/(1+exp(exp_val))
y <- rbinom(n,1,prob)
Est <- LF_logistic(X = X, y = y, loading = loading, trans = TRUE)
```

QF *Inference for quadratic forms of the regression vector in high dimensional linear regression*

Description

Computes the bias-corrected estimator of the quadratic form of the regression vector, restricted to the set of indices G for the high dimensional linear regression and the corresponding standard error. It also constructs the confidence interval for the quadratic form and test whether it is above zero or not.

Usage

```
QF(
  X,
  y,
  G,
  Cov.weight = TRUE,
  A = NULL,
  tau.vec = c(1),
  init.coef = NULL,
```

```
  lambda = NULL,
  mu = NULL,
  step = NULL,
  resol = 1.5,
  maxiter = 6,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`X`	Design matrix, of dimension n x p
`y`	Outcome vector, of length n
`G`	The set of indices, `G` in the quadratic form
`Cov.weight`	Logical, if set to `TRUE` then `A` is the $\|G\|\times\|G\|$ submatrix of the population covariance matrix corresponding to the index set `G`, else need to provide an `A` (default = TRUE)
`A`	The matrix A in the quadratic form, of dimension $\|G\|\times\|G\|$ (default = `NULL`)
`tau.vec`	The vector of enlargement factors for asymptotic variance of the bias-corrected estimator to handle super-efficiency (default = 1)
`init.coef`	Initial estimator for the regression vector (default = `NULL`)
`lambda`	The tuning parameter used in the construction of `init.coef` (default = `NULL`)
`mu`	The dual tuning parameter used in the construction of the projection direction (default = `NULL`)
`step`	The step size used to compute `mu`; if set to `NULL` it is computed to be the number of steps (< `maxiter`) to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = `NULL`)
`resol`	Resolution or the factor by which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 1.5)
`maxiter`	Maximum number of steps along which `mu` is increased/decreased to obtain the smallest `mu` such that the dual optimization problem for constructing the projection direction converges (default = 6)
`alpha`	Level of significance to test the null hypothesis which claims that the quadratic form of the regression vector is equal to 0 (default = 0.05)
`verbose`	Should inetrmediate message(s) be printed (default = `TRUE`)

Value

`prop.est`	The bias-corrected estimator of the quadratic form of the regression vector
`se`	The standard error of the bias-corrected estimator
`CI`	The matrix of confidence interval for the quadratic form of the regression vector; row corresponds to different values of `tau.vec`

`decision`	`decision`$= 1$ implies the quadratic form of the regression vector is above zero `decision`$= 0$ implies the quadratic form of the regression vector is zero row corresponds to different values of `tau.vec`
`proj`	The projection direction, of length p
`plug.in`	The plug-in estimator for the quadratic form of the regression vector restricted to `G`

References

Guo Z, Renaux C, Buhlmann P, Cai TT (2019). "Group Inference in High Dimensions with Applications to Hierarchical Testing." *Unknown*. https://arxiv.org/pdf/1909.01503.pdf.

Examples

```
n = 100
p = 200
A1gen <- function(rho,p){
A1=matrix(0,p,p)
for(i in 1:p){
 for(j in 1:p){
   A1[i,j]<-rho^(abs(i-j))
 }
}
A1
}
mu <- rep(0,p)
mu[1:5] <- c(1:5)/5
rho = 0.5
Cov <- (A1gen(rho,p))/2
beta <- rep(0,p)
beta[1:10] <- c(1:10)/5
X <- MASS::mvrnorm(n,mu,Cov)
y = X%*%beta + rnorm(n)
test.set =c(30:50)
Est <-SIHR::QF(X = X, y = y, G = test.set)
```

Index