Package: SplitKnockoff (via r-universe)

August 23, 2024

Type Package

Title Split Knockoffs for Structural Sparsity

Version 1.1

Date 2022-02-20

Author Haoxue Wang [aut, cre] (Development of the whole packages), Yang Cao [aut] (Revison of this package), Xinwei Sun [aut] (Original ideas about the package), Yuan Yao [aut] (Testing for the package and management of the development)

Maintainer Haoxue Wang <haoxwang@student.ethz.ch>

Description Split Knockoff is a data adaptive variable selection framework for controlling the (directional) false discovery rate (FDR) in structural sparsity, where variable selection on linear transformation of parameters is of concern. This proposed scheme relaxes the linear subspace constraint to its neighborhood, often known as variable splitting in optimization. Simulation experiments can be reproduced following the Vignette. We include data (both .mat and .csv format) and application with our method of Alzheimer's Disease study in this package. 'Split Knockoffs' is first defined in Cao et al. (2021) <arXiv:2103.16159>.

URL https://github.com/wanghaoxue0/SplitKnockoff

BugReports https://github.com/wanghaoxue0/SplitKnockoff/issues

Depends R (>= 3.5.0)

Imports glmnet, MASS, latex2exp, RSpectra, ggplot2, Matrix, stats, mvtnorm

Suggests knitr, rmarkdown

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

RoxygenNote 7.1.2

License MIT + file LICENSE

Date/Publication 2021-11-02 22:20:15 UTC

Repository https://wanghaoxue0.r-universe.dev

RemoteUrl https://github.com/wanghaoxue0/splitknockoff

RemoteRef HEAD

RemoteSha 259761699f55a1b4bf60fddc4c659263812d341f

Contents

`canonicalSVD` *singular value decomposition*

Description

Computes a reduced SVD without sign ambiguity

Usage

```
canonicalSVD(X)
```

Arguments

`X` the input matrix

Value

S

U

V

Examples

```
nu = 10
n = 350
m = 100
A_gamma <- rbind(matrix(0,n,m),-diag(m)/sqrt(nu))
svd.result = canonicalSVD(A_gamma)
S <- svd.result$S
S <- diag(S)
V <- svd.result$V
```

`hittingpoint` *hitting point calculator on a given path*

Description

calculate the hitting time and the sign of respective variable in a path.

Usage

```
hittingpoint(coef, lambda_vec)
```

Arguments

`coef`	the path for one variable
`lambda_vec`	respective value of lambda in the path

Value

Z: the hitting time

r: the sign of respective variable at the hitting time

`normc` *default normalization function for matrix*

Description

normalize columns of a matrix.

Usage

```
normc(X)
```

Arguments

`X`	the input martix

Value

Y the output matrix

Examples

```
library(mvtnorm)
n = 350
p = 100
Sigma = matrix(0, p, p)
X <- rmvnorm(n,matrix(0, p, 1), Sigma)
X <- normc(X)
```

---

simu_eval *functions for evaluating simulations*

Description

calculate the FDR and Power for simulations.

Usage

```
simu_eval(gamma_true, result, r)
```

Arguments

gamma_true	true signal of gamma
result	the estimated support set of gamma
r	the estimated directional effect

Value

fdr: false discovery rate of the estimated support set

power: power of the estimated support set

`simu_unit` *default unit for simulations*

Description

the simulation unit for simulation experiments.

Usage

```
simu_unit(n, p, D, A, c, k, option)
```

Arguments

`n`	the sample size
`p`	the dimension of variables
`D`	the linear transform
`A`	SNR
`c`	feature correlation
`k`	number of nonnulls in beta
`option`	option for split knockoffs

Value

simu_data: a structure contains the following elements

simu_data$fdr_split: a vector recording fdr of split knockoffs w.r.t.nu

simu_data$power_split: a vector recording power of split knockoffs w.r.t.nu

---

`sk.create` *generate split knockoff copies*

Description

Give the variable splitting design matrix and response vector. It will also create a split knockoff copy if required.

Usage

```
sk.create(X, y, D, nu, option)
```

Arguments

X	the design matrix
y	the response vector
D	the linear transform
nu	the parameter for variable splitting
option	options for creating the Knockoff copy; option$copy true : create a knockoff copy; option$eta the choice of eta for creating the split knockoff copy

Value

A_beta: the design matrix for beta after variable splitting

A_gamma: the design matrix for gamma after variable splitting

tilde_y: the response vector after variable splitting.

tilde_A_gamma: the knockoff copy of A_beta; will be [] if option$copy = false.

Examples

```
option <- array(data = NA, dim = length(data), dimnames = NULL)
option$q <- 0.2
option$eta <- 0.1
option$method <- 'knockoff'
option$normalize <- 'true'
option$lambda <- 10.^seq(0, -6, by=-0.01)
option$nu <- 10
option$copy <- 'true'
option$sign <- 'enabled'
option <- option[-1]
library(mvtnorm)
sigma <-1
p <- 100
D <- diag(p)
m <- nrow(D)
n <- 350
nu = 10
c = 0.5
Sigma = matrix(0, p, p)
for( i in 1: p){
  for(j in 1: p){
    Sigma[i, j] <- c^(abs(i - j))
 }
}
X <- rmvnorm(n,matrix(0, p, 1), Sigma)
beta_true <- matrix(0, p, 1)
varepsilon <- rnorm(n) * sqrt(sigma)
y <- X %*% beta_true + varepsilon
creat.result  <- sk.create(X, y, D, nu, option)
A_beta  <- creat.result$A_beta
A_gamma <- creat.result$A_gamma
tilde_y <- creat.result$tilde_y
```

```
tilde_A_gamma <- creat.result$tilde_A_gamma
```

---

sk.decompose *make SVD as well as orthogonal complements*

Description

make SVD as well as orthogonal complements

Usage

```
sk.decompose(X, randomize)
```

Arguments

X	the input matrix
randomize	whether to randomize

Value

U

S

V

U_perp : orthogonal complement for U

Examples

```
library(mvtnorm)
n = 350
p = 100
Sigma = matrix(0, p, p)
X <- rmvnorm(n,matrix(0, p, 1), Sigma)
decompose.result <- sk.decompose(X)
U_perp <- decompose.result$U_perp
```

`sk.filter` *Split Knockoff filter for structural sparsity*

Description

the main function, Split Knockoff filter, for variable selection in structural sparsity problem.

Usage

```
sk.filter(X, D, y, option)
```

Arguments

X	the design matrix
D	the linear transform
y	the response vector
option	various options for split knockoff filter, the details will be specified in the example

Value

results: a cell with the selected variable set in each cell w.r.t. nu.

Z: a cell with the feature significance Z in each cell w.r.t. nu.

t_Z: a cell with the knockoff significance tilde_Z in each cell w.r.t. nu.

Examples

```
option <- list(data = NA, dim = length(data), dimnames = NULL)

# the target (directional) FDR control
option$q <- 0.2

# choice on threshold, the other choice is 'knockoff+'
option$method <- 'knockoff'

# degree of separation between original design and its split knockoff copy
# in the range of [0, 2], the less the more separated
option$eta <- 0.1

# whether to normalize the dataset
option$normalize <- 'true'

# choice on the set of regularization parameters for split LASSO path
option$lambda <- 10.^seq(0, -6, by=-0.01)

# choice of nu for split knockoffs
option$nu <- 10
```

```
# choice on whether to estimate the directional effect, 'disabled'/'enabled'
option$sign <- 'enabled'

option <- option[-1]

# Settings on simulation parameters
k <- 20   # sparsity level
A <- 1    # magnitude
n <- 350  # sample size
p <- 100  # dimension of variables
c <- 0.5  # feature correlation
sigma <-1 # noise level
# generate D
D <- diag(p)
m <- nrow(D)
# generate X
Sigma = matrix(0, p, p)
for( i in 1: p){
  for(j in 1: p){
    Sigma[i, j] <- c^(abs(i - j))
  }
}
library(mvtnorm)
set.seed(100)
X <- rmvnorm(n,matrix(0, p, 1), Sigma)
# generate beta and gamma
beta_true <- matrix(0, p, 1)
for( i in 1: k){
  beta_true[i, 1] = A
  if ( i%%3 == 1){
    beta_true[i, 1] = -A
  }
}
gamma_true <- D %*% beta_true
S0 <- which(gamma_true!=0)
# generate varepsilon
set.seed(1)
# generate noise and y
varepsilon <- rnorm(n) * sqrt(sigma)
y <- X %*% beta_true + varepsilon
filter_result <- sk.filter(X, D, y, option)
Z_path <- filter_result$Z
t_Z_path <- filter_result$t_Z
```

---

sk.select *split knockoff selector given W statistics*

Description

split knockoff selector given W statistics

Usage

```
sk.select(W, q, option)
```

Arguments

`W`	statistics W_j for testing null hypothesis
`q`	target FDR
`option`	option$method can be 'knockoff' or 'knockoff+'

Value

S array of selected variable indices

`W_sign` *W statistics generator for directional FDR control*

Description

generate the split knockoff statistics W for a split LASSO path, take the directional effect into account

Usage

```
W_sign(X, D, y, nu, option)
```

Arguments

`X`	the design matrix
`D`	the linear transform
`y`	the response vector
`nu`	the parameter for variable splitting
`option`	options for creating the Knockoff statistics option$eta specify the choice of eta for creating the knockoff copy; option$lambda specify the choice of lambda for the split LASSO path

Value

W: the knockoff statistics

Z: feature significance

r: the sign estimator

t_Z: knockoff significance

W_support — *W statistics generator for FDR control*

Description

generate the split knockoff statistics W for a split LASSO path, only consider the support set estimation

Usage

```
W_support(X, D, y, nu, option)
```

Arguments

`X`	the design matrix
`D`	the linear transform
`y`	the response vector
`nu`	the parameter for variable splitting
`option`	options for creating the Split Knockoff statistics; option$eta specify the choice of eta for creating the knockoff copy; option$lambda specify the choice of lambda for the split LASSO path

Value

W: the knockoff statistics

Z: feature significance

t_Z: knockoff significance

Index