Package 'MIP'

March 2, 2018

Type Package

Title Multiple Influential Point Detection

Version 2.0

Date 2018-2-1

Author Chao Liu <muclxyliuchao@163.com>

Maintainer Lu Niu <niulu1205@buaa.edu.cn>

Description By explicitly taking into account the covariance structure of Y and the idea of random group deletion, we propose a novel procedure named MIP, short for multiple influential point detection for high-dimensional data. Along the process, we propose two novel quantities named Max and Min statistics to assess the extremeness of each point when data are subsampled. The Min statistic is useful for overcoming the swamping effect but less effective for masked influential observations, while the Max statistic is well suited for detecting masked influential observations but is less effective in handling the swamping effect. Combining their advantages, we propose a computationally efficient yet simple Min-Max algorithm for obtaining a clean subset of the data that contains no influential points.

License GPL

LazyLoad yes

Depends

NeedsCompilation no

R topics documented:

MIP-package *Multiple Influential Point Detection*

Description

This funciton is to implement the multiple influential point (MIP) detection algorithm of Zhao et al.(2016). MIP algorithm aims to detect the multiple influential observations of high dimensional space. There are two major steps: Min-Max step and Checking step. Applying the Min-Max step, an estimate of clean set is obtained. The Min-step and Max-step are implemented by the function "fun_swamping" and "fun_masking" in this package, respectively. The Min-step is used to remove the influential points of moderate or strong effect, and the following Max-step removing those of weak effect. Finally, based on the estimated clean set, one can implement the Checking step by the function "fun_checking".

Details

Package:	MIP
Type:	Package
Version:	2.0
Date:	2018-2-1
License:	GPL 2.0 or later
LazyLoad:	yes

Author(s)

Chao Liu

Maintainer: Lu Niu

fun_checking *function to check whether there are non-influential points being identified as influential ones(Checking step)*

Description

After the Min-Max step (i.e. applying function "fun_masking" and "fun_swamping" iteratively), one can get an estimate of clean set. The complementary of the estimated clean set may still contain some non-influention points. This function is to check whether some non-influential points are falsely identified as influential ones.

Usage

```
fun_checking(X, Y, n, p, q, inf_t, clean_t, alpha)
```

Arguments

X	the data of predictors with dimension n by p
Y	the data of response with dimension n by q
n	the sample size
p	the dimension of predictor
q	the dimension of response
inf_t	the estimated indices of influential poins found by Min-Max algorithm
clean_t	the estimated indices of clean poins found by Min-Max algorithm
alpha	significance level used in FDR procedure

Value

the influential points detected by the MIP algorithm

inf_setfinal	the estimated indices of influential points obtained by MIP algorithm, after applying the checking algorithm to the potential influential point inf_t.

---

fun_masking *function to detect the influential points using the Max-statistics(Max-step)*

Description

This function is to detect the influential points using the Max-statistics.

Usage

```
fun_masking(X, Y, n, p, q, n_subset, subset_vol, clean_setv, alpha)
```

Arguments

X	the data of predictors with dimension n by p
Y	the data of response with dimension n by q
n	the sample size
p	the dimension of predictor
q	the dimension of response
n_subset	the number of subsets chosen at random to compute the Min and Max statistics
subset_vol	the samples size in each subset
clean_setv	an input value of estimated clean set obtained during the iteration of Min-Max step
alpha	significance level used in FDR procedure

Value

return the size of clean set and the indices of the observations in the clean_set

S_clean	the size of the clean set
clean_set	the indices of the estimated clean set obtained by Max-step

fun_pv *function to comulate the max-statistics and the min-statistics*

Description

This functioin is to compute the Max-statistics and the Min-statistics in MIP algorithm of MIP.

Usage

```
fun_pv(X, Y, n, p, q, n_subset, subset_vol, clean_setv)
```

Arguments

X	the data of predictors with dimension n by p
Y	the data of response with dimension n by q
n	the sample size
p	the dimension of predictor
q	the dimension of response
n_subset	the number of subsets chosen at random to compute the Min and Max statistics
subset_vol	the samples size in each subset
clean_setv	the estimated clean set obtained during the iteration of Min-Max step

Value

the max-statistics and the min-statistics

T1	the values of the max-statistics
T2	the values of the min-statistics

fun_swamping *function to detect the influential points using the Min-statistics(Min-step)*

Description

Applying this function, one can remove the influential points of moderate or strong effect, alleviating the swamping effect.

Usage

```
fun_swamping(X, Y, n, p, q, n_subset, subset_vol, clean_setv, ep = 0.1, alpha)
```

Arguments

X	the data of predictors with dimension n by p
Y	the data of response with dimension n by q
n	the sample size
p	the dimension of predictor
q	the dimension of response
n_subset	the number of subsets chosen at random to compute the Min and Max statistics
subset_vol	the samples size in each subset
clean_setv	an input value of estimated clean set obtained by Min/Max step
ep	the upper bound on the proportion of rejected null hypothesis in the Min-step. The defaulted value is set at 0.1.
alpha	significance level used in FDR procedure

Value

return the clean set updated

clean_setv	the indices of the estimated clean set obtained by the min-statistics

---

MIP *function to detect multiple influential point*

Description

With predictors X and responses Y, this function is to indentify the influential points by implemeting the MIP algorithm proposed by ZHAO et al.(2016)

Usage

```
MIP(X, Y, n, p, q, n_subset, subset_vol, ep = 0.1, alpha)
```

Arguments

X	the data of predictors with dimension n by p
Y	the data of response with dimension n by q
n	the sample size
p	the dimension of predictor
q	the dimension of response
n_subset	the number of subsets chosen at random to compute the Min and Max statistics
subset_vol	the samples size in each subset
ep	the upper bound on the proportion of the rejected null hypothesis in the Min-step. The defaulted value is set at 0.1.
alpha	significance level used in FDR procedure

Details

This funciton is to implement the multiple influential point (MIP) detection algorithm of Zhao et al.(2016). MIP algorithm aims to detect the multiple influential observations of high dimensional space. There are two major steps: Min-Max step and Checking step. Applying the Min-Max step, an estimate of clean set is obtained. The Min-step and Max-step are implemented by the function "fun_swamping" and "fun_masking" in this package, respectively. The Min-step is used to remove the influential points of moderate or strong effect, and the following Max-step removing those of weak effect. Finally, based on the estimated clean set, one can implement the Checking step by the function "fun_checking".

Value

the indices of the influential points detected by the MIP algorithm

`inf_setfinal` the indices of the influential points detected by MIP algorithm

References

Zhao, J., Liu, C., Niu, L., and Leng, C. (2016). Multiple influential point detection in high-dimensional spaces. arXiv:1609.03320v2

Examples

```
#example:masking
#step 1:generating dataset, X1,Y1 represents the clean set, while X2,Y2 represents the influential set
library(MASS)
n_out=10
n=100
p=1000
q=1
n_subset=100
mx_shift=5
alpha=0.05
subset_vol=n/2
A=diag(rep(1,p))
 for (i in 1:p)
 { for (j in i:p)
 {
  A[i,j]=0.5^(abs(j-i))
  A[j,i]=A[i,j]
 }
 }
X1=mvrnorm(n,mu=rep(0,p),Sigma = A)
beta=matrix(c(0.4,0.5,0.5,0.6,0.4,rep(0,p-5)),p,1)
Y1<- X1%*%beta+rnorm(n)
X2=matrix(0,n_out,p); Y2=rep(0,n_out)
for (j in 1:n_out)
{a=sample(c(1:n),size =10,replace=FALSE,prob=NULL)
X2[j,]=X1[which(Y1==max(Y1)),]
X2[j,a]=X2[j,a]+j/1000
Y2[j]=max(Y1)+mx_shift+rnorm(1,0,0.5)*j/1000}
X=rbind(X2[1:n_out,],X1[(n_out+1):n,])  # combination of influential and non-influential observations.
Y1[1:n_out]=Y2[1:n_out]
Y=t(Y1)
#step 2: call the function "MIP" to detect the influnecial points
infset_index=MIP(X,Y,n,p,q,n_subset,subset_vol,ep=0.1,alpha)  #output rhe influential point index
```

```
print(infset_index)
```

Index