Package 'MIRL'

April 11, 2018

Type Package

Title Multiple Imputation Random Lasso for Variable Selection with Missing Entries

Version 1.0

Author Ying Liu, Yuanjia Wang, Yang Feng, Melanie M. Wall

Maintainer Ying Liu <summeryingl@gmail.com>

Description Implements a variable selection and prediction method for high-dimensional data with missing entries following the paper Liu et al. (2016) <doi:10.1214/15-AOAS899>. It deals with missingness by multiple imputation and produces a selection probability for each variable following stability selection. The user can further choose a threshold for the selection probability to select a final set of variables. The threshold can be picked by cross validation or the user can define a practical threshold for selection probability. If you find this work useful for your application, please cite the method paper.

License GPL-2

Depends glmnet,mice,MASS,boot

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-11 16:42:22 UTC

R topics documented:

mirl *mirl*

Description

This function produce the stability selection probability and the estimated coefficients

Usage

```
mirl(x=NULL,y,q2,im=5,E=NULL,lam=exp(seq(from=log(0.55),to=log(0.001),length.out=70)))
```

Arguments

x	This is the n by p design matrix with missing entries.
y	This is a n by 1 vector of the outcome, the outcome should be non missing. Please delete any sample with missing outcome.
q2	This is the number of variables to be bootstraped, recommended size is p/2.
im	Number of multiple imputation, increase of im will increase time cost. Default is 5.
E	You can use the 'mice' function in the mice package to generate this E which is a 'mids' data type, if E is entered, E will be used and x will be ignored.
lam	The vector of tunning parameter for each lasso implemented.

Value

Probabilty	This is the selection probability for each covariate. The larger the probability, the more significant the variable is related to the outcome. Notice that the probability and coef are p+1 vectors and the first coef is the intercept term, where the probability is always zero.
coef	the coefficient estimated

Author(s)

Ying Liu

References

Liu Y, Wang Y, Feng Y, Wall MM. VARIABLE SELECTION AND PREDICTION WITH INCOMPLETE HIGH-DIMENSIONAL DATA. The annals of applied statistics. 2016;10(1):418-450. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4872715/

Examples

```
#This example is a similar simulation setting in the reference paper.
cor=0.6
prob=0.02
p=10
n=200 #sample size
Sigma=matrix(cor,p,p)#correlation of predictors
diag(Sigma)=1
mu=numeric(p)
set.seed(3)
#C is the complete design matrix without missing
C=mvrnorm(n,mu,Sigma)
#The missing indicator matrix
A<-matrix(rbinom(n*p,size=1,prob),n,p)
A[,c(1,4,6)]=0 #columns without missing
p1=inv.logit(C[,1]+C[,6]-2)
A[,5]=rbinom(n,size=1,p1) #Missing at Random
p2=inv.logit(-C[,1]-0.5*C[,6]-2)
A[,10]=rbinom(n,size=1,p2)
p3=inv.logit(C[,4]-2)
A[,9]=rbinom(n,size=1,p3)
beta=numeric(p)
beta[1:6]=c(0.1,0.2,0.5,-0.3,-.4,-0.5)*5
ct=c(0,beta)
#generating Y
Y=C%*%beta+rnorm(n)
B=C
B[A==1]=NA

fit<-mirl(B,Y,p/2,im=5)
cbind(fit$coef,fit$Probability)
```

`threshold` *threshold*

Description

This function picked the best cutting threshold for the selection probability. To pick a final set of variables. The criterion is based on cross validation, to find the threshold that minimize the prediction error in validation set. Notice that this function will run slower than the MIRL function itself since it is running mirl multiple times.

Instead the user can also self define the selection probability threshold. to save the complexity of choosing by CV.

Usage

```
threshold(x, y, q2, im,m=4,thr=c(0.5,0.6,0.7,0.8,0.9))
```

Arguments

x	This is the n by p design matrix with missing entries.
y	This is a n by 1 vector of the outcome, the outcome should be non missing. Please delete any sample with missing outcome.
q2	This is the number of variables to be bootstraped, recommended size is p/2.
im	Number of multiple imputation, increase of im will increase time cost. Default is 5.
m	Number of folds for cross validation. The default is 4. Notice that increase this number will increase time linearly.
thr	The threshold grid to pick from.

Value

best	The best threhold.

Author(s)

Ying Liu

Examples

```
#This example is a simulation setting in the reference paper.
cor=0.6
prob=0.02
p=10
n=200 #sample size
Sigma=matrix(cor,p,p)#correlation of predictors
diag(Sigma)=1
mu=numeric(p)
set.seed(3)
#C is the complete design matrix without missing
C=mvrnorm(n,mu,Sigma)
#The missing indicator matrix
A<-matrix(rbinom(n*p,size=1,prob),n,p)
A[,c(1,4,6)]=0 #columns without missing
p1=inv.logit(C[,1]+C[,6]-2)
A[,5]=rbinom(n,size=1,p1) #Missing at Random
p2=inv.logit(-C[,1]-0.5*C[,6]-2)
A[,10]=rbinom(n,size=1,p2)
p3=inv.logit(C[,4]-2)
A[,9]=rbinom(n,size=1,p3)
beta=numeric(p)
beta[1:6]=c(0.1,0.2,0.5,-0.3,-.4,-0.5)*5
ct=c(0,beta)
#generating Y
Y=C%*%beta+rnorm(n)
B=C
B[A==1]=NA
```

```
best<-threshold(B,Y,q2=p/2,m=3,thr=c(0.75,0.85))
fit<-mirl(B,Y,3,p/2,im=5)
#the column number for selected variables
select=which(fit$Probability>best)
```

Index